

**OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER
MENGUNAKAN ALGORITME GENETIKA
(STUDI KASUS: SMAN 5 MALANG)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ni'mah Firsta Cahya Susilo

NIM: 145150200111137



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER MENGGUNAKAN ALGORITME
GENETIKA (STUDI KASUS: SMAN 5 MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer


Disusun Oleh :
Ni'mah Firsta Cahya Susilo
NIM: 145150200111137


Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2


Budi Darma Setiawan, S.Kom, M.Cs
NIP.198410152014041002


Agus Wahyu Widodo, S.T, M.Cs
NIP.197408052001121001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

IDENTITAS TIM PENGUJI

Informasi pelaksanaan ujian dan penguji.

JADWAL

Jadwal pelaksanaan ujian

Hari/Tanggal	SENIN / 30 JULI 2018
Tempat	FILKOM UB / F3.5
Waktu	10:30 - 11:45

PENGUJI

Majelis penguji ujian skripsi



Indriati, S.T, M.Kom (ke I) *
ketua majelis
NIP. 19831013 201504 2 002



**Imam Cholissodin, S.Si,
M.Kom (ke II)**
NIK. 201201 850719 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 30 Juli 2018



Ni'mah Firsta Cahya Susilo

NIM: 145150200111137



DAFTAR RIWAYAT HIDUP

Data Pribadi

Nama lengkap : Ni`mah Firsta Cahya Susilo
Tempat, tanggal lahir : Malang, 27 Oktober 1996
Jenis Kelamin : Perempuan
Status : Belum Kawin
Kewarganegaraan : Indonesia
Alamat : Jl Kertoraharjo 70 RT. 03 RW. 02
Kelurahan Ketawanggede
Kec. Lowokwaru, Kota Malang
No. HP : 085646625370
Email : cahyapertama27@gmail.com



Latar Belakang Pendidikan

Pendidikan Formal

2014-Sekarang : S1 Teknik Informatika Fakultas Ilmu Komputer
Universitas Brawijaya
2011-2014 : SMAN 5 Malang
2008-2011 : SMPN 4 Malang
2008-2002 : SDN Ketawanggede 1 Malang

Pengalaman Organisasi

1. Asisten Praktikum mata Kuliah Pemrograman Web TA. 2016/2017
2. Asisten Praktikum mata Kuliah Pemrograman Web TA. 2017/2018
3. Ketua Divisi Multimedia LPM DISPLAY FILKOM periode 2017/2018
4. Anggota Divisi Reportase LPM DISPLAY FILKOM periode 2016/2017
5. Anggota Divisi Reportase LPM DISPLAY FILKOM periode 2015/2016

UCAPAN TERIMA KASIH

Penelitian ini tentunya tidak dapat terselesaikan tanpa dukungan dari berbagai pihak. Penulis secara khusus mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penulisan skripsi ini. Pada kesempatan ini penulis menyampaikan rasa terima kasih kepada:

1. Allah SWT dengan segala rahmat dan karunia-Nya yang telah memberikan kekuatan bagi penulis untuk menyelesaikan skripsi ini,
2. Orang tua tercinta Ayahanda Susilo Mansurudin dan Ibunda Nuraini Susanti yang selama ini telah membantu penulis dengan memberikan dukungan dalam bentuk perhatian, kasih sayang, semangat, serta doa yang tak henti-hentinya demi kelancaran dan terselesaikannya skripsi ini. Adik M. Hamzah Al-Kautsar B. dan M. Nauval Al-Aizar R. yang juga telah memberikan dukungan pada penulis,
3. Bapak Budi Darma Setiawan, S.Kom, M.Cs selaku pembimbing satu yang mengajarkan kedisiplinan dalam menyelesaikan skripsi ini serta telah menjadi pembimbing yang sabar dan solutif dalam memberikan bimbingan kepada penulis,
4. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku pembimbing dua yang telah membimbing yang sabar dan solutif dalam memberikan bimbingan bagi penulis,
5. Bapak Zahid, S.Kom selaku narasumber yang telah meluangkan waktunya dan memberikan pengarahan pada skripsi ini,
6. Teman-teman program studi Teknik Informatika, Fakultas Ilmu Komputer yang tidak dapat penulis sebutkan satu per satu yang telah memberikan dukungan moril, menjadi teman dan sahabat yang baik selama menempuh studi.

Semoga Allah SWT senantiasa membalas semua kontribusi dan bantuan yang telah diberikan pada penulis. Tidak lupa penulis sampaikan bahwa penelitian ini masih memiliki banyak kekurangan. Diskusi mengenai penelitian selanjutnya yang membangun merupakan beberapa hal yang dapat disampaikan pada penulis.

ABSTRACT

Ni`mah Firsta Cahya Susilo, Optimasi Penjadwalan Ujian Akhir Semester Menggunakan Algoritme Genetika (Studi Kasus: SMAN 5 Malang)

Supervisors: Budi Darma Setiawan, S.Kom, M.Cs and Agus Wahyu Widodo, S.T, M.Cs

Scheduling is an activity with detailed division of time for various purposes in various fields. Scheduling of the semester final exam at SMAN 5 Malang in practice there are still some obstacles such as limited examination supervisors and the distribution of the subjects tested is less suitable. SMAN 5 Malang has applied the exam schedule by dividing subjects based on national examination subjects and other subjects. This study aims to determine the effect of parameter changes on genetic algorithms and find a scheduling solution for the semester final exam at SMAN 5 Malang with a genetic algorithm. Scheduling the final semester exam in this study is divided into subject scheduling and scheduling of exam supervisors. The testing of individual subjects and supervisors is carried out 5 times with population size of 100, number of generations 500 and combination of crossover rate 0.5 and mutation rate 0.5. Testing results in an optimal population for 180 subjects while for supervisors a total of 150. In testing the combination of Cr and Mr values found a combination of optimal values for individual subjects is Cr = 0.7 and Mr = 0.3 and a combination of Cr = 0.6 and Mr = 0.4 for individual supervisors with an optimal number of generations in individual subjects is 200 generations while in individual supervisors is 400 generations.

Keyword: optimization, scheduling, genetic algorithm, final exam of semester

ABSTRAK

Ni'mah Firsta Cahya Susilo, Optimasi Penjadwalan Ujian Akhir Semester Menggunakan Algoritme Genetika (Studi Kasus: SMAN 5 Malang)

Dosen Pembimbing: Budi Darma Setiawan, S.Kom, M.Cs dan Agus Wahyu Widodo, S.T, M.Cs

Penjadwalan merupakan suatu kegiatan dengan pembagian waktu yang rinci untuk berbagai tujuan pada berbagai bidang. Penjadwalan ujian akhir semester di SMAN 5 Malang dalam praktiknya masih ditemukan beberapa kendala seperti terbatasnya pengawas ujian serta distribusi mata pelajaran yang diujikan kurang sesuai. SMAN 5 Malang telah menerapkan jadwal ujian dengan membagi mata pelajaran berdasarkan mata pelajaran ujian nasional dan mata pelajaran lainnya. Penelitian ini bertujuan untuk mengetahui dampak perubahan parameter pada algoritme genetika dan menemukan solusi penjadwalan ujian akhir semester pada SMAN 5 Malang dengan algoritme genetika. Penjadwalan ujian akhir semester dalam penelitian ini terbagi menjadi penjadwalan mata pelajaran dan penjadwalan pengawas ujian. Pengujian individu mata pelajaran dan pengawas dilakukan sebanyak 5 kali dengan ukuran populasi sebesar 100, jumlah generasi sebesar 500 serta kombinasi nilai *crossover rate* sebesar 0.5 dan *mutation rate* sebesar 0.5. Pengujian menghasilkan jumlah populasi optimal untuk individu mata pelajaran sejumlah 180 sedangkan untuk individu pengawas sejumlah 150. Pada pengujian kombinasi nilai Cr dan Mr ditemukan kombinasi nilai optimal untuk individu mata pelajaran adalah Cr = 0.7 dan Mr = 0.3 dan kombinasi nilai Cr = 0.6 dan Mr = 0.4 untuk individu pengawas dengan jumlah generasi optimal pada individu mata pelajaran adalah 200 generasi sedangkan pada individu pengawas adalah 400 generasi.

Kata kunci: optimasi, penjadwalan, algoritme genetika, ujian akhir semester

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Optimasi Penjadwalan Ujian Akhir Semester Menggunakan Algoritme Genetika (Studi Kasus: SMAN 5 Malang)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Budi Darma Setiawan, S.Kom, M.Cs dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Bayu Priyambadha, S.Kom, M.Kom selaku Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika,
4. Ibu Candra Dewi, S.Kom, M.Sc selaku dosen Penasihat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
5. Bapak Zahid, S.Kom selaku narasumber yang telah meluangkan waktunya dan memberikan pengarahan pada skripsi ini,
6. Ayahanda Susilo Mansurudin dan Ibunda Nuraini Susanti atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
7. Adik penulis M. Hamzah Al-Kautsar B. dan M. Nauval Al-Aizar R. serta keluarga besar penulis yang telah memberikan semangat yang tak henti-hentinya untuk segera menyelesaikan skripsi ini,
8. Sahabat Dayinta Warih W. dan Rhevitta Widyaning P. yang memberikan semangat dan telah berjuang bersama selama menempuh studi,
9. Rekan-rekan LPM DISPLAY FILKOM yang telah memberikan semangat, memberikan dukungan dan pengalaman selama tiga periode,
10. Teman-teman Teknik Informatika angkatan 2014 yang telah memberikan semangat,
11. Seluruh civitas academica Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan.

Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 30 Juli 2018

Penulis

cahyapertama27@gmail.com



DAFTAR ISI

PENGESAHAN	ii
IDENTITAS TIM PENGUJI	iii
PERNYATAAN ORISINALITAS	iv
DAFTAR RIWAYAT HIDUP	v
UCAPAN TERIMA KASIH	vi
ABSTRACT	vii
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Optimasi	6
2.3 Penjadwalan	6
2.4 Algoritme Genetika	6
2.4.1 Representasi Kromosom	7
2.4.2 Inisialisasi Populasi	7
2.4.3 Evaluasi	7
2.4.4 Operator Genetika	8
BAB 3 METODOLOGI PENELITIAN	10
3.1 Metodologi Penelitian	10

3.2 Studi Literatur	10
3.3 Pengumpulan Data	11
3.4 Perancangan	11
3.5 Implementasi	12
3.6 Pengujian	12
3.7 Pengambilan Kesimpulan.....	12
BAB 4 PERANCANGAN.....	13
4.1 Formulasi Permasalahan.....	13
4.2 Representasi Kromosom.....	14
4.3 Tahapan Penyelesaian Masalah dengan Algoritme Genetika	17
4.3.1 Representasi Kromosom	18
4.3.2 Crossover.....	19
4.3.3 Mutasi	21
4.3.4 Seleksi.....	23
4.4 Perhitungan Manual	25
4.4.1 Inisialisasi Populasi Awal	25
4.4.2 Crossover.....	30
4.4.3 Mutasi	34
4.4.4 Seleksi.....	37
4.5 Perancangan Pengujian	38
4.5.1 Perancangan Pengujian Ukuran Populasi	38
4.5.2 Perancangan Pengujian Kombinasi Crossover Rate (Cr) dan Mutation Rate (Mr)	38
BAB 5 IMPLEMENTASI	40
5.1 Lingkungan Sistem	40
5.1.1 Lingkungan Perangkat Keras	40
5.1.2 Lingkungan Perangkat Lunak	40
5.2 Implementasi Algoritme	40
5.2.1 Implementasi Inisialisasi Populasi Awal	40
5.2.2 Implementasi Crossover	41
5.2.3 Implementasi Mutasi	44
5.2.4 Implementasi Constraint.....	44

5.2.5 Implementasi Perhitungan Fitness	68
5.2.6 Implementasi Seleksi.....	68
5.3 Implementasi Antar Muka	69
BAB 6 PENGUJIAN DAN ANALISIS.....	74
6.1 Hasil Pengujian Terhadap Ukuran Populasi	74
6.2 Hasil Pengujian Terhadap Kombinasi Crossover rate (Cr) dan Mutation Rate (Mr).....	77
6.3 Hasil Pengujian Terhadap Jumlah Generasi.....	81
6.4 Analisis Global.....	83
BAB 7 PENUTUP	85
7.1 Kesimpulan.....	85
7.2 Saran	85
DAFTAR PUSTAKA.....	86
LAMPIRAN A DATA MATA PELAJARAN.....	88
LAMPIRAN B DATA GURU PENGAWAS	92
LAMPIRAN C HASIL JADWAL MATA PELAJARAN UJIAN AKHIR SEMESTER.....	98
LAMPIRAN D HASIL JADWAL PENGAWAS UJIAN AKHIR SEMESTER	102

DAFTAR TABEL

Tabel 2.1 Kajian pustaka	4
Tabel 2.2 Representasi kromosom.....	7
Tabel 2.3 Inisialisasi populasi	7
Tabel 3.1 Kebutuhan data	11
Tabel 4.1 Batasan dan penalti kromosom mata pelajaran	13
Tabel 4.2 Batasan dan bobot penalti kromosom pengawas.....	13
Tabel 4.3 Pembagian waktu sesi ujian akhir semester	14
Tabel 4.4 Pengkodean mata pelajaran.....	15
Tabel 4.5 Pengkodean pengawas ujian.....	15
Tabel 4.6 Representasi kromosom mata pelajaran	16
Tabel 4.7 Representasi kromosom pengawas	16
Tabel 4.8 Populasi awal mata pelajaran	26
Tabel 4.9 Populasi awal pengawas.....	27
Tabel 4.10 Proses <i>multi-point crossover</i>	30
Tabel 4.11 Proses <i>one-cut-point crossover</i>	31
Tabel 4.12 Proses <i>random exchange mutation</i> mata pelajaran	34
Tabel 4.13 Proses <i>random exchange mutation</i> pengawas	35
Tabel 4.14 Hasil perhitungan <i>fitness</i> individu mata pelajaran.....	37
Tabel 4.15 Hasil seleksi individu mata pelajaran	37
Tabel 4.16 Hasil perhitungan <i>fitness</i> individu pengawas.....	37
Tabel 4.17 Hasil seleksi individu mata pengawas	37
Tabel 4.18 Perancangan pengujian ukuran populasi	38
Tabel 4.19 Perancangan pengujian kombinasi Cr dan Mr	38
Tabel 5.1 Lingkungan perangkat keras.....	40
Tabel 5.2 Lingkungan perangkat lunak	40
Tabel 5.3 Implementasi kode program populasi awal mata pelajaran	40
Tabel 5.4 Implementasi kode program inisialisasi populasi awal pengawas	41
Tabel 5.5 Implementasi kode program <i>multi-point crossover</i>	42
Tabel 5.6 Implementasi kode program <i>one-cut-point crossover</i>	43
Tabel 5.7 Implementasi kode program <i>random exchange mutation</i>	44

Tabel 5.8 Implementasi kode program constraint mata pelajaran 1	45
Tabel 5.9 Implementasi kode program constraint mata pelajaran 2	45
Tabel 5.10 Implementasi kode program constraint mata pelajaran 3	48
Tabel 5.11 Implementasi kode program constraint mata pelajaran 4	52
Tabel 5.12 Implementasi kode program constraint pengawas 1	53
Tabel 5.13 Implementasi kode program constraint pengawas 2	55
Tabel 5.14 Implementasi kode program constraint pengawas 3	67
Tabel 5.15 Implementasi kode program perhitungan <i>fitness</i>	68
Tabel 5.16 Implementasi kode program <i>elitism selection</i>	68
Tabel 6.1 Pengujian ukuran populasi pada individu mata pelajaran.....	75
Tabel 6.2 Pengujian ukuran populasi pada individu pengawas.....	76
Tabel 6.3 Hasil pengujian kombinasi Cr dan Mr pada individu mata pelajaran ...	79
Tabel 6.4 Hasil pengujian kombinasi Cr dan Mr pada individu pengawas	80
Tabel 6.5 Detail pelanggaran <i>constraint</i> individu mata pelajaran.....	83
Tabel 6.6 Detail pelanggaran <i>constraint</i> individu pengawas.....	84

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi <i>elitism selection</i>	8
Gambar 2.2 Ilustrasi <i>crossover</i>	8
Gambar 2.3 Ilustrasi <i>random exchange mutation</i>	9
Gambar 3.1 Diagram blok metodologi penelitian	10
Gambar 4.1 Representasi kromosom	14
Gambar 4.2 Diagram alir algoritme genetika	18
Gambar 4.3 Diagram alir inisialisasi populasi awal	19
Gambar 4.4 Diagram alir <i>one-cut-point crossover</i>	21
Gambar 4.5 Diagram alir <i>random exchange mutation</i>	23
Gambar 4.6 Diagram alir <i>elitism selection</i>	25
Gambar 5.1 Antarmuka halaman input	69
Gambar 5.2 Antarmuka halaman proses	70
Gambar 5.3 Halaman data mata pelajaran	70
Gambar 5.4 Halaman data pengawas	71
Gambar 5.5 Halaman hasil mata pelajaran	71
Gambar 5.6 Halaman hasil pengawas	72
Gambar 5.7 Halaman hasil jadwal mata pelajaran	72
Gambar 5.8 Halaman hasil jadwal pengawas	73
Gambar 6.1 Grafik pengujian ukuran populasi mata pelajaran	77
Gambar 6.2 Grafik pengujian ukuran populasi pengawas	77
Gambar 6.3 Grafik pengujian kombinasi Nilai Cr dan Mr individu mata pelajaran	81
Gambar 6.4 Grafik hasil pengujian kombinasi nilai Cr dan Mr individu pengawas	81
Gambar 6.5 Grafik pengujian jumlah generasi mata pelajaran	82
Gambar 6.6 Grafik pengujian jumlah generasi individu pengawas	83

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Jadwal berdasarkan Kamus Besar Bahasa Indonesia (2003) merupakan suatu pembagian waktu terhadap rencana pengurutan suatu kegiatan dengan pembagian waktu pelaksanaan yang rinci. Dengan adanya jadwal, kegiatan yang akan dilakukan untuk berbagai tujuan seperti penjadwalan pada dunia pendidikan, transportasi maupun industri. Penjadwalan merupakan suatu aktivitas yang umum dilakukan dalam dunia pendidikan misalnya pada sekolah, universitas ataupun tempat kursus. Penjadwalan digunakan dalam segala macam aspek kegiatan pendidikan, misalnya pada penyusunan jadwal mata pelajaran, jadwal ujian tengah semester, jadwal pengawas ujian ataupun jadwal ujian akhir semester.

Begitu pula dengan SMA Negeri 5 Malang pada beberapa periode tertentu yakni akhir semester selalu melakukan penjadwalan ujian akhir semester. SMA Negeri 5 Malang merupakan sekolah menengah atas yang memiliki 30 kelas yang terdiri dari tiga jurusan untuk masing-masing kelas X, XI, dan XII yakni jurusan MIPA, IPS dan bahasa. Dalam praktiknya masih terdapat kendala yang terjadi dalam penjadwalan. Beberapa kendala yang sering dijumpai yakni keterbatasan pengawas ujian, bentrok jadwal pengawas ujian serta distribusi mata pelajaran yang kurang sesuai. Adapun jangka waktu pelaksanaan ujian akhir semester yang terlalu lama karena disebabkan oleh pembagian jadwal mata pelajaran yang kurang efisien. Selain penjadwalan mata pelajaran yang diujikan, pada penjadwalan ujian akhir semester di SMA Negeri 5 Malang juga dilakukan penjadwalan pengawas ujian. Penjadwalan mata pelajaran dilakukan berdasarkan bobot tingkat kesulitan yang dimiliki tiap mata pelajaran dan tiap kelas. Bobot tersebut digunakan untuk pemerataan tingkat kesulitan mata pelajaran pada satu hari. Sedangkan untuk penjadwalan pengawas ujian akan dilakukan berdasarkan mata pelajaran yang diajarkan serta bentrok antar pengawas juga menjadi batasan dalam penjadwalan pengawas. Penjadwalan pengawas ujian akan dilakukan setelah jadwal mata pelajaran telah selesai dilakukan.

Dalam dunia ilmu komputer permasalahan penjadwalan tersebut dapat dilakukan optimasi dengan beberapa cara. Metode optimasi yang dapat digunakan adalah metode heuristik yang merupakan metode pencarian berdasarkan aturan-aturan empiris untuk memperoleh solusi yang lebih baik dari sebelumnya. Metode heuristik juga terdiri dari *evolutionary computing* dan algoritme *local search* yang bertujuan untuk menemukan solusi dari sebuah permasalahan (Mahmudy, 2013). Pada penelitian "*A Genetic Algorithm To Solve The Timetable Problem*" membandingkan optimasi penjadwalan menggunakan algoritme genetika yang termasuk *evolutionary computing* dengan *tabu search* yang termasuk algoritme *local search*, hasil dari penelitian tersebut yakni algoritme genetika memberikan solusi penjadwalan yang lebih baik apabila dibandingkan dengan solusi yang dihasilkan oleh *tabu search* (Colorni, et al., 1992). Beberapa penelitian tentang

pemanfaatan algoritme genetika dalam optimasi penjadwalan banyak dilakukan salah satunya adalah penelitian "*Genetic Algorithm for Exam Proctor*" yang membahas mengenai penyelesaian masalah penjadwalan pengawas ujian dengan algoritme genetika (Promcharoen, et al., 2009).

Algoritme genetika merupakan salah satu algoritme yang biasanya digunakan untuk menyelesaikan permasalahan optimasi. Algoritme genetika adalah algoritme pencarian stokastik yang berdasarkan mekanisme genetika dan seleksi alam. Algoritme ini telah menghasilkan solusi pencarian global yang optimum dan efisien (Holland, 1975). Goldbreg (1975) menyatakan bahwa cara kerja algoritme genetika menyerupai proses biologi dari reproduksi dan seleksi alam untuk menghasilkan solusi yang terbaik. Tiga hal utama yang terdapat pada algoritme ini adalah operator seleksi, *mutasi* dan *crossover*. Pada proses seleksi akan memilih induk untuk generasi selanjutnya. Operasi mutasi dilakukan pada individu induk untuk menghasilkan anak. Proses selanjutnya adalah menggabungkan induk dan anak untuk generasi selanjutnya. Individu dengan *fitness* yang baik akan menjadi generasi selanjutnya (Ram & Davim, 2017).

Berdasarkan latar belakang yang telah diuraikan sebelumnya, penulis mengangkat suatu penelitian yang berjudul "Optimasi Penjadwalan Ujian Akhir Semester Menggunakan Algoritme Genetika (Studi Kasus: SMAN 5 Malang). Diharapkan dengan algoritme genetika dapat menyelesaikan permasalahan yang sebelumnya terjadi serta menghasilkan solusi jadwal ujian akhir semester yang optimal.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka dapat dirumuskan masalah yang akan diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana pengaruh perubahan parameter pada algoritme genetika pada optimasi penjadwalan ujian akhir semester di SMAN 5 Malang?
2. Bagaimana kualitas dari solusi penjadwalan ujian akhir semester dengan menggunakan algoritme genetika?

1.3 Tujuan

Tujuan yang akan dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui dampak dari perubahan parameter algoritme genetika pada optimasi penjadwalan ujian akhir semester di SMAN 5 Malang.
2. Mengetahui kualitas dari solusi dari hasil penjadwalan ujian akhir semester dengan menggunakan algoritme genetika.

1.4 Manfaat

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Memberikan kemudahan pada bagian kurikulum sekolah dalam penyusunan jadwal ujian akhir semester di SMAN 5 Malang.

2. Meningkatkan efisiensi dalam penyusunan jadwal ujian akhir semester di SMAN 5 Malang.
3. Memberikan solusi penjadwalan yang akan diterapkan pada ujian akhir semester.

1.5 Batasan Masalah

Berdasarkan penjelasan latar belakang dan rumusan masalah yang telah dijelaskan sebelumnya, batasan-batasan masalah yang ada dalam penelitian ini diantaranya adalah sebagai berikut:

1. Objek yang digunakan dalam penelitian ini adalah data penjadwalan ujian akhir semester pada SMAN 5 Malang.
2. Data dalam penjadwalan yang digunakan yakni data mata pelajaran yang ada untuk masing-masing kelas di SMAN 5 Malang.

1.6 Sistematika Pembahasan

Penelitian ini disusun secara sistematis sehingga akan terlihat keterkaitan antar masing-masing bab. Adapun isi dari masing-masing bab sebagai berikut:

1. BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang penelitian, rumusan masalah, tujuan, manfaat penelitian, batasan masalah serta sistematika laporan.

2. BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi tentang kajian pustaka yang sesuai dengan karakteristik penelitian serta landasan teori yang diperoleh dari berbagai sumber pustaka.

3. BAB 3 METODOLOGI

Pada bab ini berisi mengenai metode penelitian yang dipilih untuk menyelesaikan permasalahan pada penelitian.

4. BAB 4 PERANCANGAN

Pada bab ini berisi mengenai perancangan yang dilakukan untuk menyelesaikan permasalahan pada penelitian.

5. BAB 5 IMPLEMENTASI

Pada bab ini berisi mengenai hasil implementasi dari perancangan yang telah dilakukan sebelumnya pada penelitian untuk menyelesaikan permasalahan penelitian.

6. BAB 6 PENUTUP

Pada bab ini berisi mengenai ringkasan atau kesimpulan dari capaian penelitian serta saran pengembangan dari penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini akan berfokus pada permasalahan penyusunan jadwal ujian akhir semester pada SMA Negeri 5 Malang. Dalam sistem yang akan dibangun dengan algoritme genetika ini akan terdapat beberapa *constraint* yang hanya dapat diterapkan pada penjadwalan ujian akhir semester di SMA Negeri 5 Malang. Representasi kromosom pada sistem ini akan terdiri dari mata pelajaran, sesi, pengawas ujian kelas serta jurusan yang ada. Keluaran yang akan dihasilkan berupa beberapa solusi terbaik untuk penyusunan jadwal dengan nilai *fitness* terbaik.

Pada penelitian sebelumnya yang berjudul “*Comparing Heuristic, Evolutionary and Local Search Approaches to Scheduling*” telah menghasilkan solusi jadwal produksi dengan algoritme genetika yang lebih baik apabila dibandingkan dengan algoritme *local search* dan algoritme pencarian heuristik. Perbedaan yang terdapat pada penelitian ini adalah penggunaan algoritme genetika dengan representasi kromosom, jenis mutasi, jenis *crossover* serta jenis seleksi yang berbeda dengan penelitian sebelumnya. Beberapa pustaka yang digunakan sebagai referensi pada penelitian ini dirangkum dalam tabel 2.1.

Tabel 2.1 Kajian pustaka

No	Judul	Objek	Metode	Hasil
1	Implementasi Algoritma Genetika Untuk Penjadwalan <i>Customer Service</i> (Studi Kasus: Biro Perjalanan Kangaroo)	Jadwal <i>customer service</i> Biro Perjalanan Kangaroo	Algoritme genetika	Penelitian ini menghasilkan ukuran populasi 110, ukuran generasi 110 dan perbandingan <i>crossover rate</i> dan <i>mutation rate</i> 0.7:0.3.
2	<i>Formulation of Genetic Algorithm to Generate Good Quality Course Timetable</i>	Jadwal mata kuliah pada Universitas Devi Ahilya	Algoritme genetika	Penelitian ini menghasilkan jadwal kuliah untuk setiap semester yang ada di universitas.
3	<i>Genetic algorithm and heuristic search for solving timetable problem case</i>	Jadwal mata kuliah Universitas Pelita Harapan	Algoritme genetika dan pencarian heuristik	Penelitian ini menghasilkan bahwa kedua algoritme berhasil menyelesaikan

	<i>study: Universitas Pelita Harapan timetable</i>			permasalahan penjadwalan
4	<i>Genetic Algorithm for Exam Proctor Schedule</i>	Jadwal pegawai ujian	Algoritme genetika	Penelitian ini menghasilkan parameter genetika dengan jumlah populasi 40, Cr sebesar 0.9, Mr sebesar 0.1 dan generasi maksimum sebanyak 2000 menghasilkan nilai <i>fitness</i> yang lebih baik.
5	<i>Comparing Heuristic, Evolutionary and Local Search Approaches to Scheduling</i>	Jadwal produksi pada gudang	<i>Local search</i> , pencarian heuristik dan algoritme genetika	Penelitian ini menghasilkan bahwa dari ketiga algoritme memberikan solusi yang terbaik apabila dibandingkan dengan dua algoritme lainnya.
6	Implementasi Algoritma genetika untuk Optimasi Penggunaan Lahan Pertanian	Lahan pertanian	Algoritme genetika	Penelitian ini menghasilkan bahwa nilai parameter algoritme genetika berpengaruh terhadap hasil optimasi yang dihasilkan.
7	<i>A Two-level Genetic Algorithm for the Multi-Mode Resource-Constrained Project Scheduling Problem</i>	<i>Multimode resource-constrained project</i>	Algoritme genetika	Penelitian ini menghasilkan jumlah populasi optimal sebanyak 5 kali jumlah aktivitas dengan nilai Mr sebesar 0.001 dan jumlah generasi optimasi sebanyak 50 generasi.

Sumber: (Damayanti, et al., 2017), (Jain, et al., 2010), (Lukas, et al., 2009), (Promcharoen, et al., 2009), (Rana, et al., 1996), (Saputro, et al., 2015), dan (Magalhães-Mendes, 2011)

2.2 Optimasi

Optimasi merupakan suatu usaha untuk menemukan kegunaan yang paling efektif atau paling baik dalam suatu situasi ataupun sumber daya. Permasalahan optimasi merupakan sebuah masalah komputasi dimana bertujuan untuk menemukan solusi terbaik dari solusi yang mungkin ada (Black, 2001). Permasalahan optimasi dapat diselesaikan dengan menggunakan metode *evolutionary computing* dan algoritme *local search*.

2.3 Penjadwalan

Penjadwalan merupakan sebuah rencana tertulis yang bertujuan untuk mengalokasikan urutan dan waktu untuk setiap operasi yang dibutuhkan untuk diselesaikan. Berdasarkan Kamus Besar Bahasa Indonesia (2003) jadwal adalah suatu pembagian waktu terhadap rencana pengurutan suatu kegiatan dengan pembagian waktu pelaksanaan yang rinci. Dari definisi yang ada mengenai penjadwalan dapat disimpulkan bahwa penjadwalan merupakan sebuah kegiatan dengan pembagian waktu tertentu yang bertujuan untuk menyelesaikan sebuah operasi. Penjadwalan dilakukan untuk berbagai tujuan dan juga dapat diterapkan pada bidang pendidikan, transportasi maupun industri. Dalam dunia pendidikan, (Schaerf, 1999) melakukan klasifikasi jadwal yang ada sebagai berikut:

1. Jadwal mata pelajaran

Jadwal ini merupakan jadwal mingguan untuk semua kelas di sekolah. Batasan yang ada biasanya tidak adanya bentrok guru untuk dua kelas di jam yang sama.

2. Jadwal mata kuliah

Jadwal ini merupakan jadwal mingguan untuk semua dosen dalam suatu universitas.

3. Jadwal ujian

Jadwal ini merupakan jadwal ujian mata yang menghindari bentrok terhadap mata kuliah atau mata pelajaran.

2.4 Algoritme Genetika

Algoritme evolusi telah banyak dikembangkan mulai dari tahun 1960-an untuk menyelesaikan berbagai masalah optimasi yang dirasa kompleks dan sulit untuk diselesaikan. Algoritme evolusi yang banyak digunakan untuk menyelesaikan permasalahan ini yaitu algoritme genetika (Gen & Cheng, 1999). Algoritme genetika merupakan algoritme pencarian berdasarkan prinsip evolusi yang diamati di alam. Algoritme ini menggabungkan operator genetika seperti seleksi, *crossover* dan mutasi dengan tujuan untuk mendapatkan solusi terbaik yang memungkinkan untuk suatu permasalahan. Pencarian algoritme genetika akan terus berjalan sampai kriteria *termination* telah terpenuhi (Goldberg, 1989).

2.4.1 Representasi Kromosom

Tahap pertama dalam algoritme genetika merupakan representasi kromosom. Kromosom terbentuk dari gen-gen yang telah dipetakan dari permasalahan penjadwalan seperti pada Tabel 2.2. Representasi kromosom yang digunakan pada penelitian ini adalah representasi permutasi.

Tabel 2.2 Representasi kromosom

Kromosom	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5
1
2

2.4.2 Inisialisasi Populasi

Kromosom yang telah terbentuk sebelumnya akan membentuk sebuah populasi. Populasi pada algoritme genetika terbentuk dari kromosom-kromosom yang telah dipetakan dari permasalahan penjadwalan. Populasi yang ditampilkan pada Tabel 2.3 terdiri dari beberapa kromosom (T).

Tabel 2.3 Inisialisasi populasi

Populasi	Kromosom					
1	T3	T2	T1
2	T5	T6	T4
3	T7	T8	T9

2.4.3 Evaluasi

Fitness dalam dunia biologi merupakan sebuah kualitas untuk mengukur efesiensi dari kromosom (Goldberg, 1989). Evaluasi pada algoritme genetika dilakukan dengan menghitung nilai *fitness* untuk masing-masing populasi. Fungsi *fitness* untuk setiap kromosom dihitung berdasarkan *constraint* yang ada dan *penalty* yang dilanggar. Pada persamaan (2.1) dimana fungsi *fitness* $f(t)$ diperoleh dari pembagian dengan pembilang 1 dan penyebut $1+\sum \text{penalty}$ dimana $\sum \text{penalty}$ merupakan jumlah penalti yang telah dilanggar dengan t merupakan individu indeks ke- t .

$$f(t) = \frac{1}{1+\sum \text{penalty}} \quad (2.1)$$

Nilai *fitness* yang dihasilkan akan berupa bilangan positif dan akan mencapai nilai maksimum 1 apabila tidak melanggar *constraint* apapun. Wood dan Trenaman (1999) mengelompokkan *constraint* menjadi dua yaitu:

1. *Hard constraint*

Hard constraint merupakan *constraint* yang harus dipenuhi dan tidak bisa dilanggar pada penjadwalan. Oleh karena itu, *hard constraint* biasanya memiliki nilai *penalty* yang lebih besar apabila dilanggar.

2. *Soft constraint*

Berbeda dengan *hard constraint*, *soft constraint* merupakan *constraint* yang tidak terlalu diperlukan dalam penjadwalan, tetapi dibutuhkan untuk menghasilkan jadwal yang baik.

2.4.4 Operator Genetika

Operator genetika pada algoritme genetika adalah sebagai berikut:

1. Seleksi

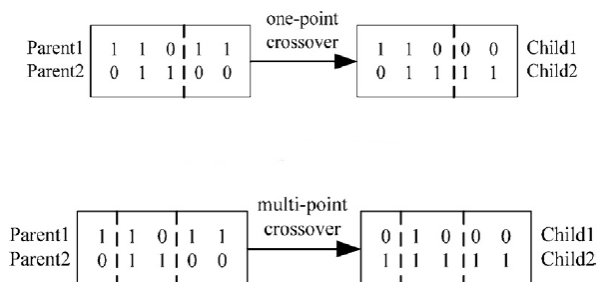
Seleksi merupakan operator yang memilih kromosom dalam populasi untuk dilakukan reproduksi. Pada penelitian ini akan digunakan *elitism selection*. *Elitism selection* akan memilih populasi dengan nilai *fitness* terbaik untuk generasi berikutnya. Pada Gambar 2.1 dilakukan seleksi untuk memilih populasi untuk generasi selanjutnya dengan mengurutkan nilai *fitness*.

Populasi	Fitness		Populasi	Fitness
P1	0.003	→	P2	0.5
P2	0.5		P3	0.09
P3	0.09		P1	0.003

Gambar 2.1 Ilustrasi *elitism selection*

2. *Crossover*

Crossover dalam algoritme genetika merupakan operator yang melakukan persilangan dan menghasilkan individu untuk generasi selanjutnya. Pada penelitian ini akan digunakan *multi-point crossover* untuk melakukan persilangan kromosom mata pelajaran dan *one-cut-point crossover* pada kromosom pengawas. Pada Gambar 2.2 *offspring* dihasilkan dari proses persilangan yang dilakukan dengan memilih titik secara acak.



Gambar 2.2 Ilustrasi *crossover*

Sumber: (Yuan, et al., 2013)

3. Mutasi

Mutasi merupakan operasi yang digunakan untuk mempertahankan keberagaman dari populasi satu generasi ke generasi selanjutnya. Mutasi yang akan digunakan pada penelitian ini adalah *random exchange mutation*. Mutasi jenis ini akan memilih dua titik kromosom secara random pada satu populasi yang akan ditukarkan. Gen ke-3 dan gen ke-7 dilakukan mutasi sehingga menghasilkan *offspring* mutasi seperti pada Gambar 2.3.

Parent	5	2	9	11	8	23	76	14	34	27
Child	5	2	76	11	8	23	9	14	34	27

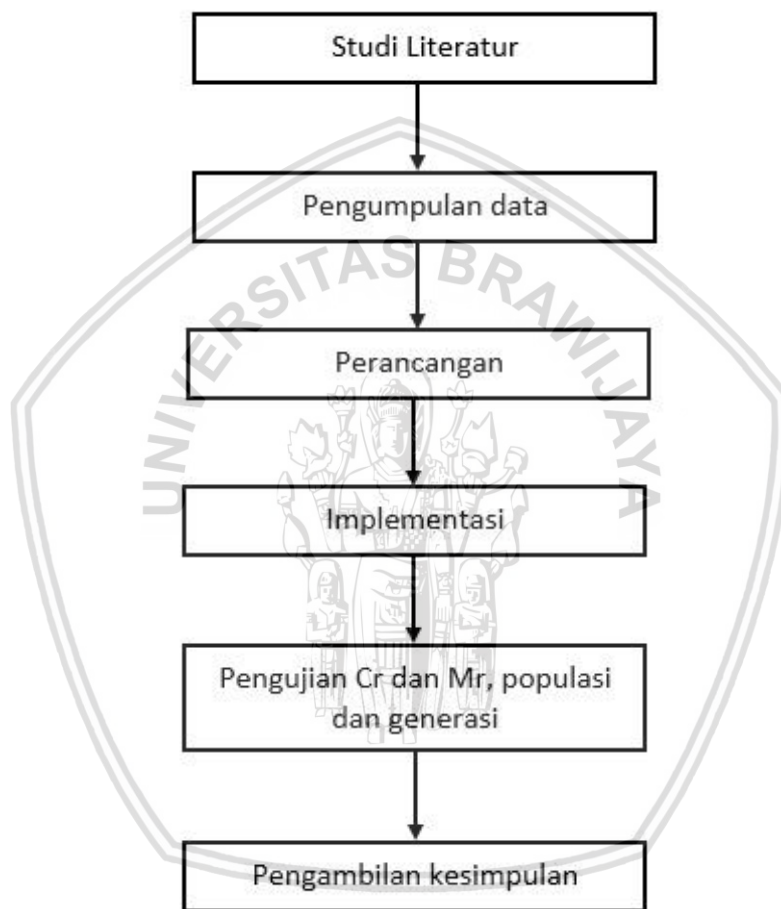
Gambar 2.3 Ilustrasi *random exchange mutation*



BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan yaitu studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi, pengujian, serta pengambilan kesimpulan. Tahapan-tahapan penelitian ini dapat diilustrasikan dalam diagram blok metodologi penelitian seperti pada gambar 3.1.



Gambar 3.1 Diagram blok metodologi penelitian

3.2 Studi Literatur

Pada tahapan studi literatur diperoleh informasi tambahan yang akan digunakan sebagai acuan pada penelitian. Literatur yang digunakan merupakan literature dari berbagai bidang yang berhubungan serta dapat mendukung optimasi jadwal ujian akhir semester, diantaranya:

1. Algoritme genetika
2. Dokumen penjadwalan ujian akhir semester SMAN 5 Malang
3. Dokumen mata pelajaran SMAN 5 Malang

Literatur-literatur tersebut diperoleh dari buku, jurnal, artikel dan dokumentasi proyek.

3.3 Pengumpulan Data

Pada tahap pengumpulan data ini variabel penelitian yang digunakan merupakan data jadwal ujian akhir semester, data mata pelajaran yang ada untuk masing-masing kelas keminatan di SMAN 5 Malang serta beberapa batasan atau *constraint* yang digunakan. Data-data yang dibutuhkan untuk penelitian ini diperoleh dari penelusuran halaman web SMAN 5 Malang yang menyediakan informasi berupa data mata pelajaran serta wawancara kepada pihak sekolah terkait *constraint* yang akan digunakan untuk menyusun jadwal ujian akhir semester. Data bobot mata pelajaran diperoleh dengan membagikan kuisisioner kepada siswa-siswa SMAN 5 Malang sebanyak 100 kuisisioner. Kuisisioner tersebut berisi tingkat kesulitan siswa terhadap satu mata pelajaran yang akan digunakan sebagai bobot mata pelajaran. Kebutuhan data-data yang dibutuhkan dalam penelitian ini ditampilkan pada Tabel 3.1.

Tabel 3.1 Kebutuhan data

No	Data kebutuhan	Bentuk data	Sumber data	Metode
1	Data mata pelajaran dan data guru-guru	Kualitatif	SMAN 5 Malang	Observasi
2	Data jumlah kelas	Kuantitatif	SMAN 5 Malang	Observasi
3	Data <i>constraint</i>	Kualitatif	SMAN 5 Malang	Wawancara
4	Data bobot mata pelajaran	Kuantitatif	Kuisisioner	Kuisisioner

3.4 Perancangan

Tahap perancangan sistem dilakukan setelah melakukan analisis terhadap kebutuhan sistem yang akan dibangun. Tahapan dalam perancangan meliputi *input*, proses dan *output*. *Input* yang akan digunakan dalam penelitian ini merupakan data-data mata pelajaran yang ada pada SMAN 5 Malang untuk masing-masing kelas keminatan. Tahap proses yang akan dilakukan pada penelitian ini meliputi representasi kromosom, pembuatan populasi awal, proses reproduksi, proses seleksi, serta proses evaluasi untuk memperoleh individu dengan nilai *fitness* terbaik. *Output* yang akan dihasilkan dari penelitian ini merupakan susunan jadwal ujian akhir semester dengan beberapa solusi jadwal yang optimal.

3.5 Implementasi

Implementasi merupakan tindakan penerapan dari tahapan perancangan yang telah disusun serta dilakukan sebelumnya. Pada tahap implementasi ini akan diterapkan komponen-komponen yang berkaitan dengan penelitian dan juga antarmuka yang telah dirancang sebelumnya. Kebutuhan sistem yang akan digunakan pada penelitian ini berupa *personal computer* dengan spesifikasi sistem operasi Windows 8.1 Pro 64 bit, Microsoft Office 2013, Netbeans IDE 8.0.2, Intel Core i3, VGA Intel HD, RAM 6.00 GB, dan *hard disk* 500 GB.

3.6 Pengujian

Tahapan pengujian dilakukan untuk mengetahui parameter pada algoritme genetika untuk menghasilkan individu dengan nilai *fitness* yang baik. Pada penelitian ini, pengujian yang akan dilakukan meliputi beberapa parameter genetika sebagai berikut:

1. Pengujian ukuran populasi
2. Pengujian kombinasi nilai Cr dan Mr
3. Pengujian jumlah generasi

3.7 Pengambilan Kesimpulan

Tahapan terakhir yang dilakukan dalam penelitian ini yakni tahap pengambilan kesimpulan. Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi serta pengujian terhadap metode yang digunakan telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang diterapkan dalam penelitian ini.

BAB 4 PERANCANGAN

4.1 Formulasi Permasalahan

Pada penelitian ini dilakukan optimasi terhadap jadwal ujian akhir semester di SMA Negeri 5 Malang yang bertujuan untuk menyelesaikan permasalahan penjadwalan yang ada dengan algoritme genetika. Penyusunan jadwal ujian yang ada masih dilakukan secara manual dengan kombinasi mata pelajaran dan pengawas setiap harinya yang masih secara acak. Hal ini menyebabkan beberapa kesulitan seperti jadwal pengawas yang bentrok, kombinasi mata pelajaran yang kurang optimal untuk siswa ataupun belum meratanya pembagian jadwal untuk mengawas bagi para guru.

Untuk mengatasi permasalahan yang telah dipaparkan sebelumnya, penelitian ini akan membangun sebuah sistem optimasi penjadwalan ujian akhir semester dengan menggunakan algoritme genetika sehingga menghasilkan solusi penjadwalan mata pelajaran dan penjadwalan pengawas yang lebih optimal. Penjadwalan ujian akhir semester yang dilakukan akan dibagi dua menjadi penjadwalan mata pelajaran dan penjadwalan pengawas. Penjadwalan pengawas sendiri akan dilakukan setelah penjadwalan mata pelajaran selesai dibuat. Penjadwalan tersebut masing-masing memiliki beberapa batasan-batasan yang akan mempengaruhi nilai *fitness* apabila batasan tersebut dilanggar. Semakin besar nilai *fitness* yang dimiliki menunjukkan semakin baik suatu individu. Pada Tabel 4.1 menampilkan batasan yang dimiliki oleh kromosom mata pelajaran yang terdiri dari tiga *hard constraint* dan satu *soft constraint* dengan bobot masing-masing. Sedangkan pada kromosom pengawas akan memiliki batasan yang berbeda dengan *hard constraint* sejumlah dua dan *soft constraint* sejumlah dua yang ditampilkan pada Tabel 4.2.

Tabel 4.1 Batasan dan penalti kromosom mata pelajaran

No	Constraint	Penalti
1	Terdapat bentrok mata pelajaran dalam satu hari	0.286
2	Terdapat mata pelajaran dalam satu hari yang memiliki bobot lebih besar daripada 6.29	0.286
3	Terdapat mata pelajaran sesi 2 memiliki bobot lebih besar daripada sesi 1	0.286
4	Terdapat mata pelajaran ujian nasional pada sesi 2	0.142
Jumlah		1

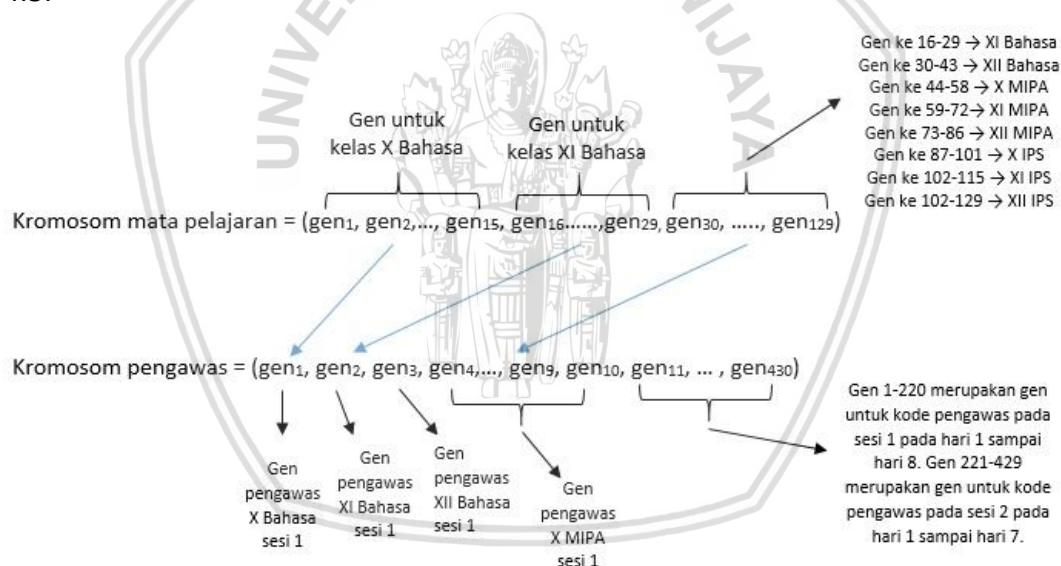
Tabel 4.2 Batasan dan bobot penalti kromosom pengawas

No	Constraint	Penalti
1	Terdapat bentrok jadwal mengawas pada tiap sesi	0.5

2	Pengawas tidak mengawas sesuai mata pelajaran yang dikuasai	0.25
3	Terdapat guru senior mengawas di sesi ke 2	0.25
Jumlah		1

4.2 Representasi Kromosom

Representasi kromosom yang digunakan dalam penelitian ini terbagi menjadi dua kromosom. Kromosom pertama merepresentasikan kromosom mata pelajaran, sedangkan kromosom kedua merepresentasikan kromosom pengawas ujian akhir semester yang merujuk pada kromosom pertama. Kromosom pertama memiliki panjang 129, sedangkan kromosom kedua memiliki panjang 430 yang ditunjukkan pada Gambar 4.1. Pada kromosom mata pelajaran yang memiliki panjang 129 terdiri dari kombinasi mata pelajaran tiap kelas dan jurusan dalam 8 hari. Hal ini disebabkan penjadwalan SMA Negeri 5 Malang memiliki aturan penjadwalan ujian dua mata pelajaran untuk satu harinya sedangkan untuk mata pelajaran yang ada untuk tiap kelas terdiri dari 15 sampai 16 mata pelajaran yang dibagi menjadi dua sesi untuk setiap harinya seperti yang ditunjukkan pada Tabel 4.3.



Gambar 4.1 Representasi kromosom

Tabel 4.3 Pembagian waktu sesi ujian akhir semester

Sesi	Jam
1	07.00 - 08.30
2	08.30 - 09.30

Pada penelitian ini, terdapat 129 kode mata pelajaran sebagai gen kromosom mata pelajaran dan 232 kode guru sebagai gen kromosom pengawas. Dalam data pengkodean mata pelajaran terdapat kode mata pelajaran, kelas,

jurusan, nama mata pelajaran, keterangan serta bobot. Sedangkan untuk pengkodean guru sebagai pengawas terdapat kode pengawas, nama, keminatan serta keterangan. Pengkodean mata pelajaran dan pengawas ditunjukkan pada Tabel 4.8. Pengkodean mata pelajaran dan pengawas yang lebih detail dapat dilihat pada LAMPIRAN A dan LAMPIRAN B.

Tabel 4.4 Pengkodean mata pelajaran

No	Kode	Kelas	Jurusan	Mata Pelajaran	Ket	Bobot
1	M1	X	Bahasa	Antropologi	UN	3.28
2	M2	X	Bahasa	Bahasa Daerah	Non-UN	3.28
3	M3	X	Bahasa	Bahasa Indonesia	UN	2.71
4	M4	X	Bahasa	Bahasa Inggris	UN	2.28
5	M5	X	Bahasa	Jerman-Jepang	UN	2.42
..
128	M128	XII	IPS	Sosiologi	UN	2.71
129	M129	XII	IPS	Wirausaha	Non-UN	2.85

Tabel 4.5 Pengkodean pengawas ujian

No	Kode Pengawas	Nama Pengawas	Keminatan	Keterangan
1	G1	Abdul Ghofir	M50	Junior
2	G1	Abdul Ghofir	M64	Junior
3	G2	Agung Setyawanto, S.Pd	M3	Junior
4	G2	Agung Setyawanto, S.Pd	M88	Junior
...
231	G66	Zahid, S.Kom	M7	Junior
232	G66	Zahid, S.Kom	M93	Junior

Representasi kromosom mata pelajaran terdiri dari 129 gen penyusun dengan 15 gen pertama berisi indeks mata pelajaran pada kelas X Bahasa, 14 gen berisi indeks mata pelajaran pada kelas XI Bahasa, 14 gen berisi indeks mata pelajaran pada kelas XII Bahasa, 15 gen berisi indeks mata pelajaran kelas X MIPA, 14 gen berisi indeks mata pelajaran kelas XI MIPA, 14 gen berisi indeks mata pelajaran kelas XII MIPA, 15 gen berisi indeks mata pelajaran kelas X IPS, 1 gen berisi indeks mata pelajaran kelas XI IPS, serta 15 gen berisi indeks mata pelajaran kelas XII IPS. Pada Tabel 4.10 ditunjukkan contoh representasi kromosom mata pelajaran.

Tabel 4.6 Representasi kromosom mata pelajaran

X Bahasa				...	XI MIPA						...	XII IPS					
H1		...	H8	...	H1		...	H8		...	H1		...	H8			
S1	S2	...	S1	...	S1	S2	...	S1	S2	...	S1	S2	...	S1	S2		
1	6	...	15	...	59	65	...	70	61	...	116	123	...	125	120		
5	10	...	4	...	62	71	...	68	60	...	129	119	...	122	127		

Berdasarkan Tabel 4.10, terdapat angka 1 pada kolom X Bahasa H1 S1. Hal tersebut menunjukkan pada mata pelajaran yang akan diujikan pada Hari ke-1 sesi 1 kelas X Bahasa adalah antropologi dengan bobot 3.28. Kode tersebut juga berlaku pada gen-gen selanjutnya yang dibangkitkan secara acak.

Representasi kromosom terdiri dari 430 gen penyusun dengan 3 gen pertama berisi indeks pengawas untuk kelas bahasa, 18 gen selanjutnya berisi indeks pengawas untuk kelas MIPA, dan 9 gen selanjutnya berisi indeks pengawas untuk kelas IPS. Jumlah dan urutan gen tersebut juga berlaku pada hari selanjutnya dan sesi 2. Pada tabel 4.11 ditunjukkan contoh representasi kromosom pengawas. Jumlah 430 gen pada kromosom pengawas didapatkan dari jumlah perkalian tiap kelas dan jumlah mata pelajaran yang ada. Jumlah mata pelajaran kelas X terdiri dari 15 mata pelajaran serta 14 mata pelajaran untuk kelas XI dan kelas XII.

Jumlah gen pengawas

$$\begin{aligned}
 &= (\text{bahasa} * (\text{mapel10} + \text{mapel11} + \text{mapel12})) \\
 &+ (\text{mipa} * (\text{mapel10} + \text{mapel11} + \text{mapel12})) \\
 &+ (\text{ips} * (\text{mapel10} + \text{mapel11} + \text{mapel12}))
 \end{aligned}$$

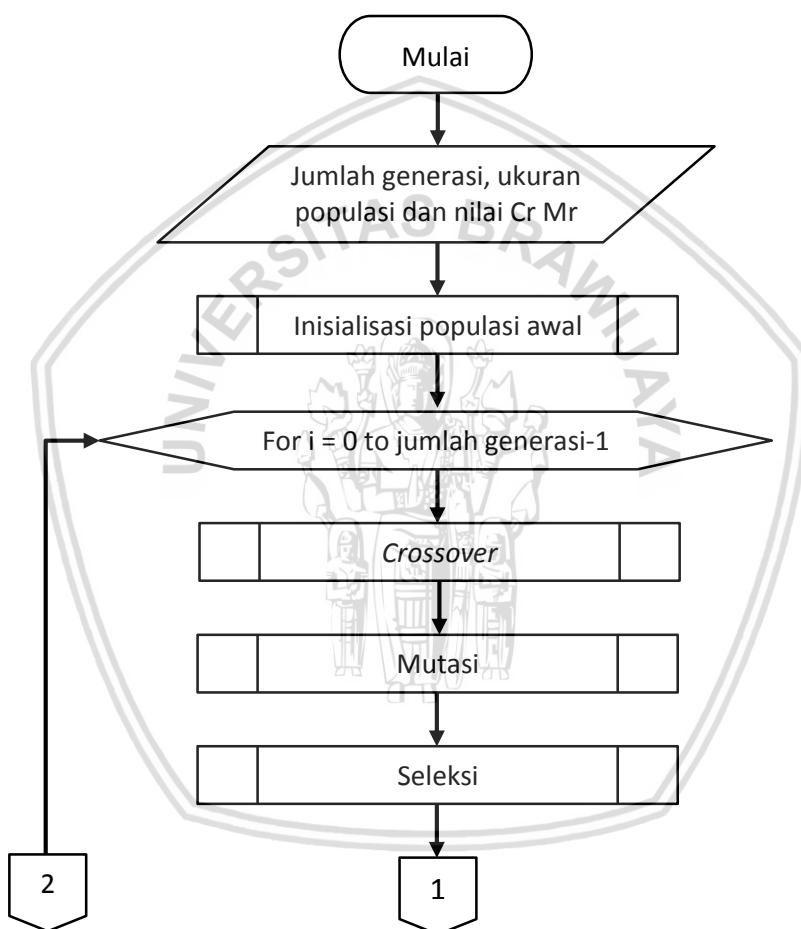
Tabel 4.7 Representasi kromosom pengawas

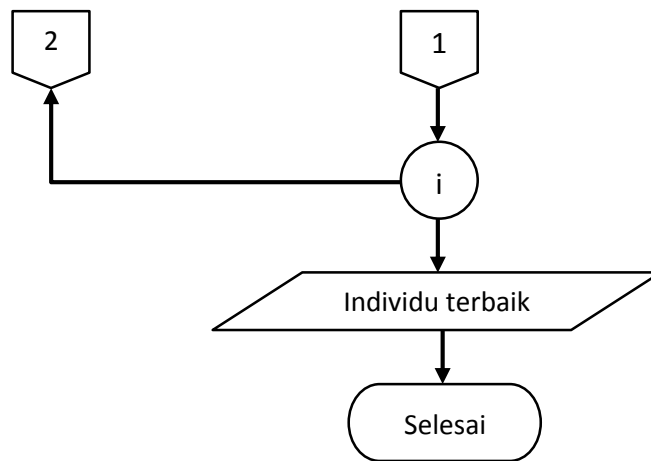
Sesi 1							Sesi 2						
Hari 1			...	Hari 7			Hari 1			...	Hari 7		
Bahasa			...	IPS			MIPA			...	IPS		
X	XI	XII	...	X	XI	XII	X	XI	XII	...	X	XI	XII
1	78	24	...	34	54	94	67	95	84	...	25	14	45
47	34	80	...	64	29	25	62	26	69	...	70	35	58

Berdasarkan Tabel 4.11, terdapat angka 1 pada kolom Sesi 1 Hari 1 Bahasa X. Hal tersebut menunjukkan pada pengawas pada sesi pertama hari ke-1 untuk jurusan Bahasa kelas X adalah Abdul Ghofir. Kode tersebut juga berlaku pada gen-gen selanjutnya yang dibangkitkan secara acak.

4.3 Tahapan Penyelesaian Masalah dengan Algoritme Genetika

Penyelesaian optimasi penjadwalan ujian akhir semester dengan menggunakan algoritme genetika pada penelitian ini melalui beberapa tahapan yang ditampilkan pada Gambar 4.1. Proses pertama yang dilakukan setelah memberi masukan jumlah populasi, generasi serta kombinasi Cr dan Mr adalah inialisasi populasi awal sesuai dengan jumlah populasi yang telah ditentukan. Tahapan berikutnya dilakukan perulangan proses reproduksi yang meliputi seleksi, *crossover* dan mutasi sebanyak generasi yang ditentukan. Proses reproduksi yang dilakukan akan menghasilkan individu terbaik yang akan melalui proses seleksi hingga menghasilkan individu yang optimal.

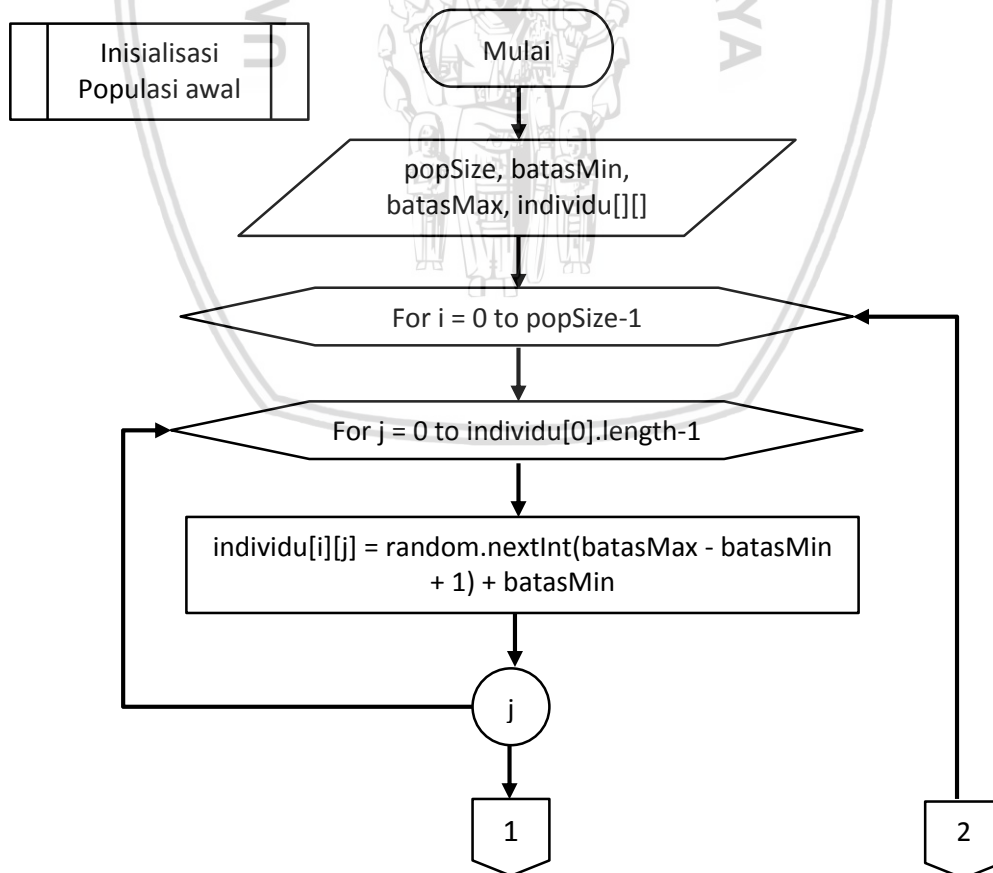


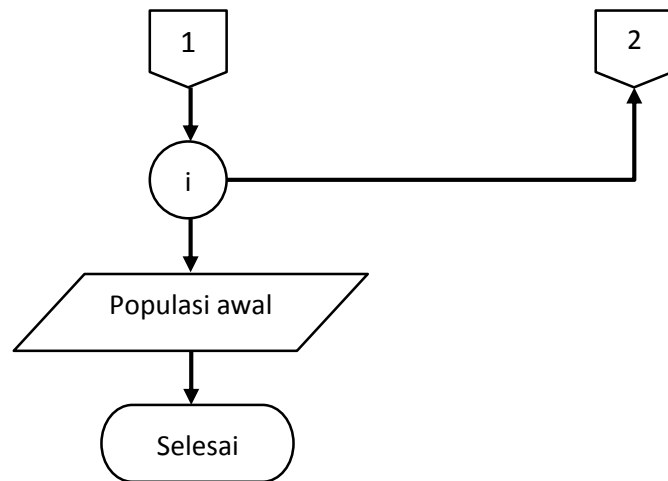


Gambar 4.2 Diagram alir algoritme genetika

4.3.1 Representasi Kromosom

Proses *generate* populasi awal ditampilkan pada Gambar 4.2. Secara garis besar proses *generate* kromosom mata pelajaran dan kromosom pengawas memiliki persamaan yang membedakan adalah nilai dari variabel batasMax dan batasMin. Kedua variabel tersebut akan menentukan batas minimum dan batas maksimal nilai random yang akan dibangkitkan untuk masing-masing kromosom.

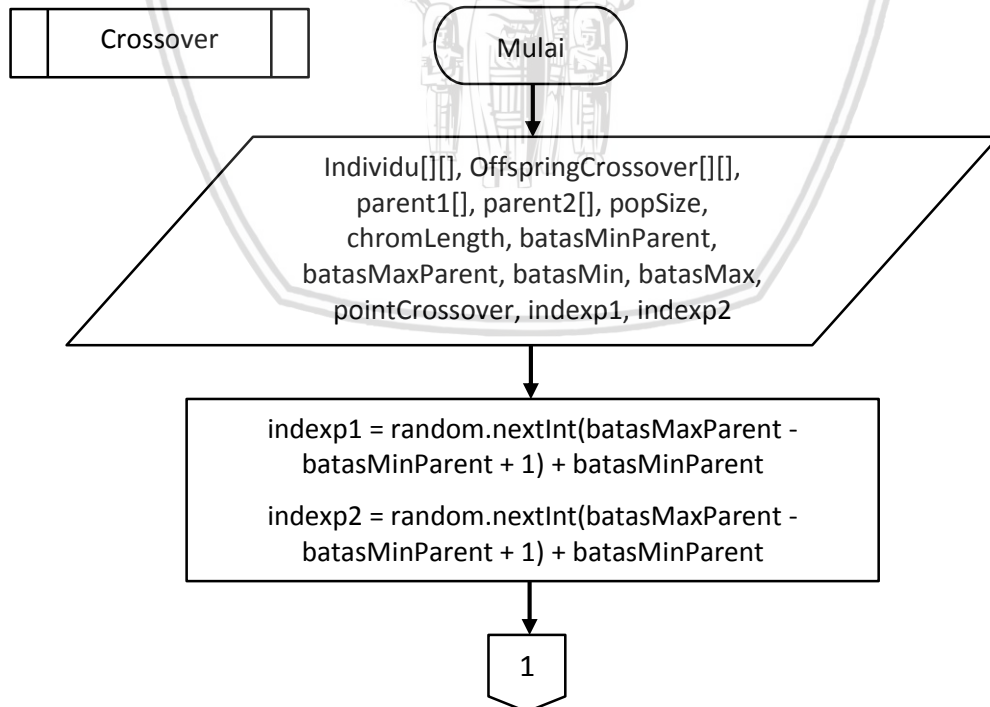


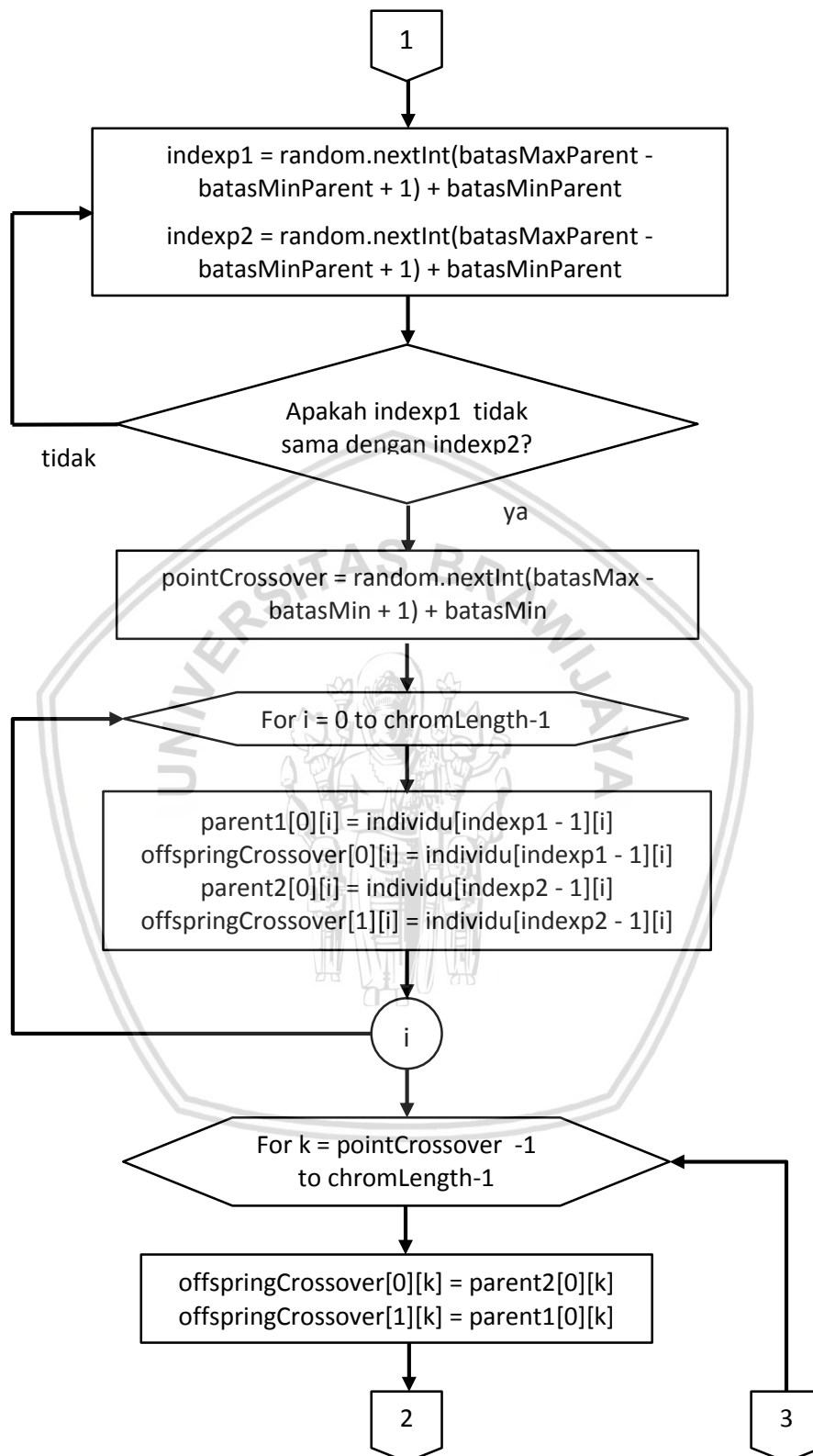


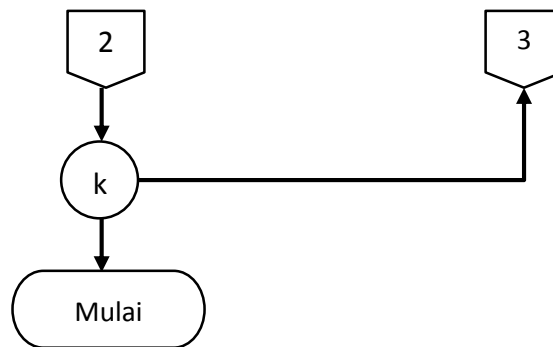
Gambar 4.3 Diagram alir inisialisasi populasi awal

4.3.2 Crossover

Pada kromosom mata pelajaran *crossover* yang digunakan adalah *multi-point crossover* sedangkan untuk proses *crossover* kromosom pengawas digunakan proses *one-cut-point crossover*. Kedua proses *crossover* ini secara garis besar memiliki persamaan. Pada Gambar 4.3 ditunjukkan proses *one-cut-point crossover*. Proses pindah silang *one-cut-point crossover* yang dilakukan dimulai dengan memilih satu titik secara acak dari individu. *Parent* yang akan dilakukan pindah silang juga dipilih secara acak.







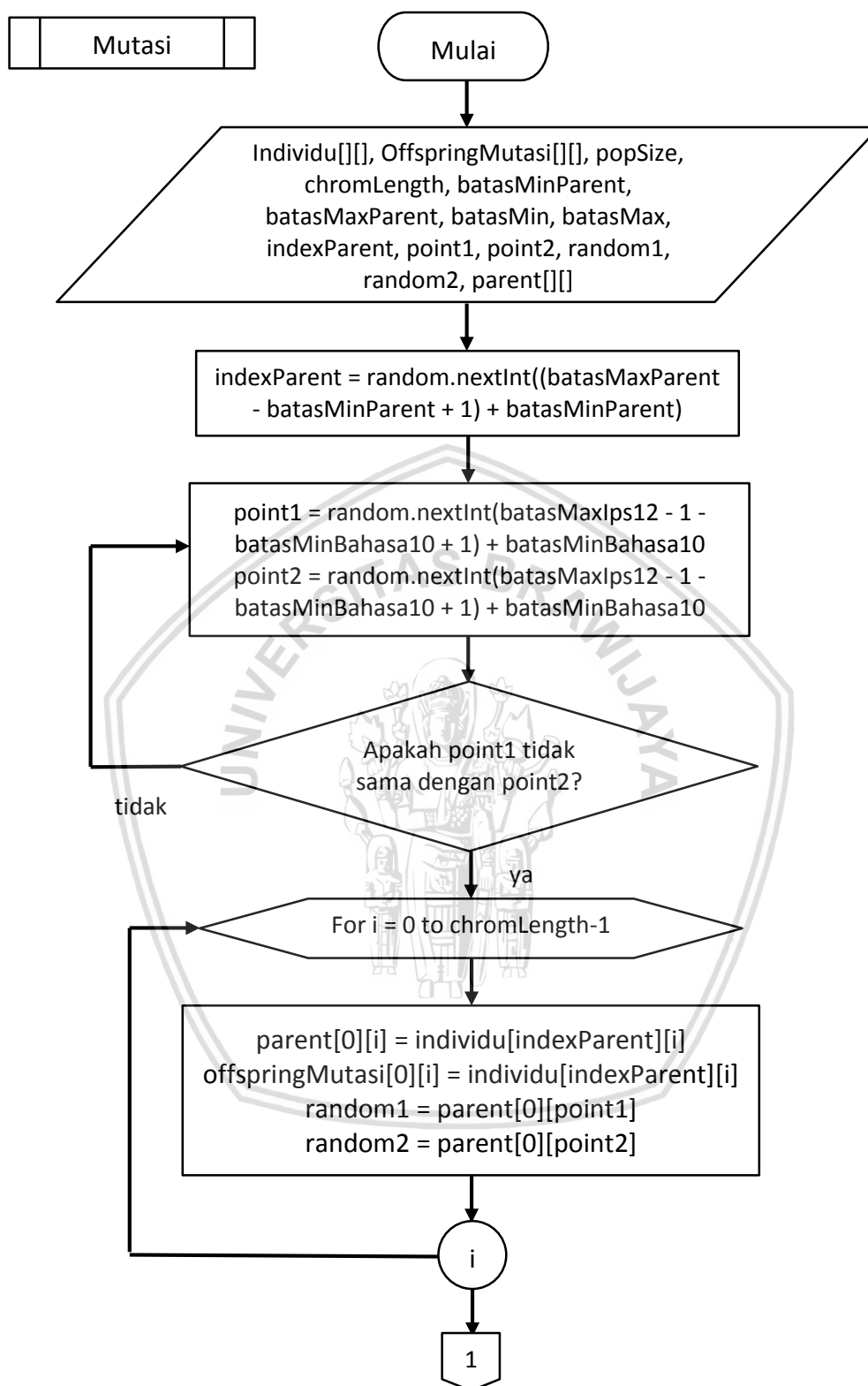
Gambar 4.4 Diagram alir *one-cut-point crossover*

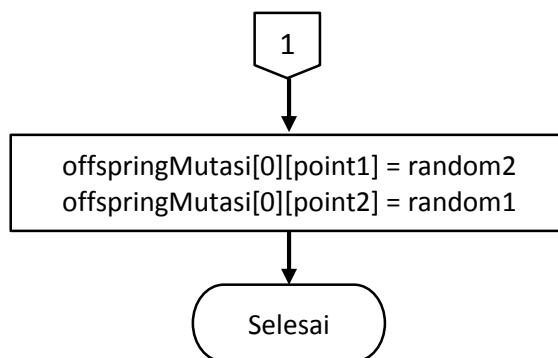
Langkah-langkah *one-cut-point crossover* yang ditunjukkan pada Gambar 4.3 adalah sebagai berikut:

1. Sistem menerima masukan kromosom berupa array individu[][] yang akan dilakukan *crossover*.
2. Sistem melakukan proses pemilihan dua parent secara random pada individu[][] sebagai induk *crossover*. Apabila induk1 sama dengan induk2 maka proses pemilihan *parent* akan dilakukan sekali lagi.
3. Sistem melakukan proses pemilihan titik *crossover* secara acak dengan batasMin sebagai batas minimal dan batasMax sebagai batas maksimal.
4. Indeks *parent* yang telah dipilih sebelumnya akan disimpan pada array parent1 dan parent2.
5. Proses pindah silang dilakukan dengan menukar dua *parent* sesuai dengan titik *crossover* yang telah dipilih.

4.3.3 Mutasi

Proses *random exchange mutation* yang dilakukan dimulai dari proses memilih dua posisi pada individu yang akan dimutasi. Proses pemilihan parent mutasi dilakukan secara acak dari individu awal serta *offspring* hasil *crossover*. Nilai dari individu dengan posisi yang telah diperoleh akan ditukarkan satu sama lain sehingga menghasilkan *offspring* mutasi. Proses-proses tersebut ditampilkan pada Gambar 4.4.





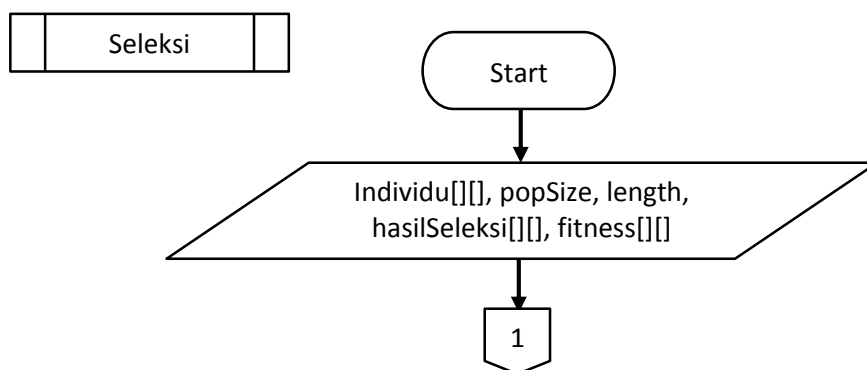
Gambar 4.5 Diagram alir *random exchange mutation*

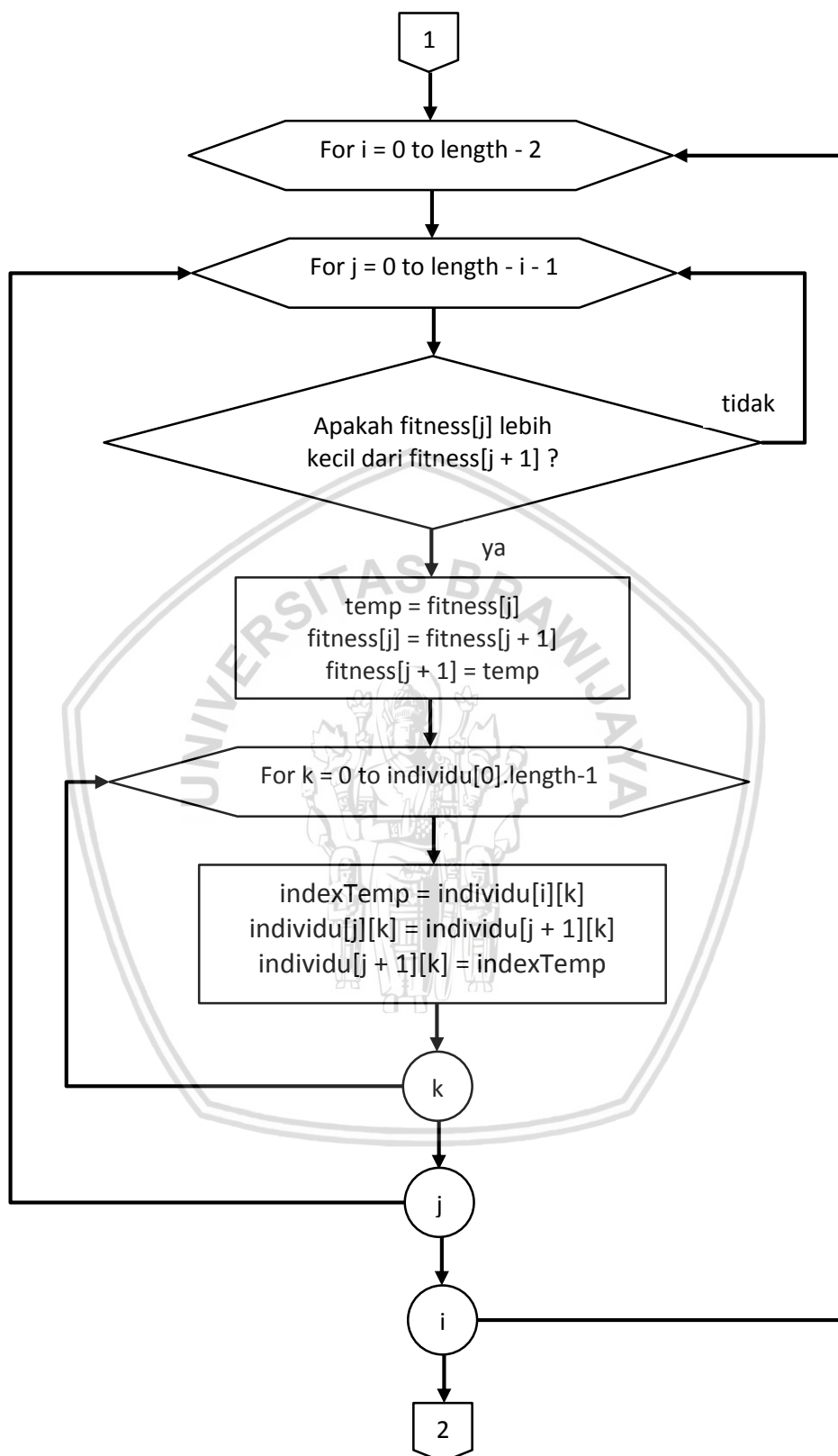
Langkah-langkah proses mutasi *random exchange mutation* yang ditunjukkan pada Gambar 4.4 adalah sebagai berikut:

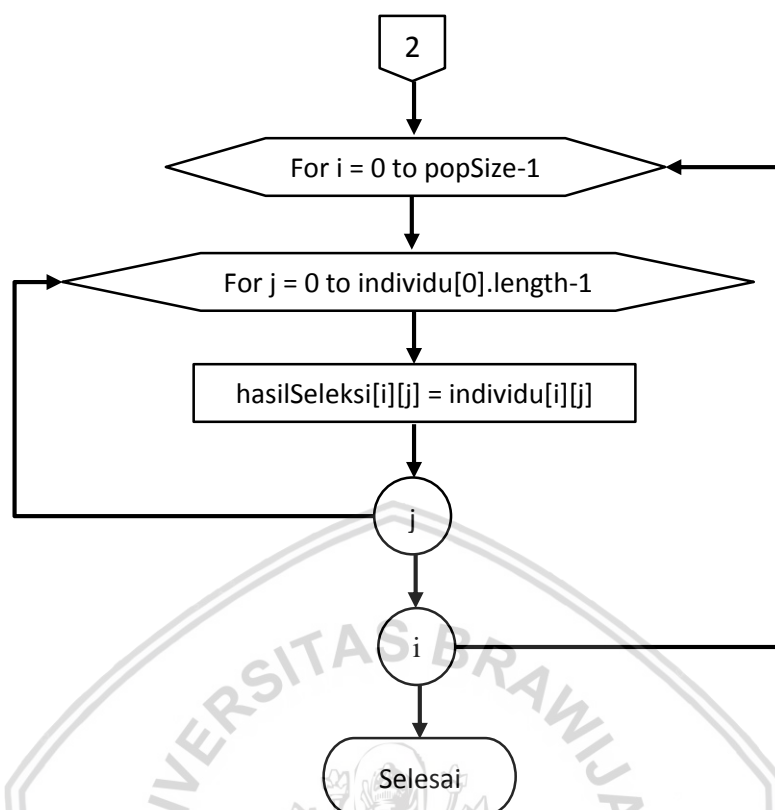
1. Sistem menerima masukan kromosom berupa array individu[][] yang akan dilakukan mutasi.
2. Sistem melakukan proses pemilihan parent mutasi yang dipilih secara random berdasarkan batasMinParent sebagai batas minimal dan batasMaxParent sebagai batas maksimal.
3. Pemilihan dua titik mutasi akan dipilih secara random dengan batas minimal dan maksimal yang ditentukan. Apabila kedua titik mutasi yaitu point1 dan point2 memiliki nilai yang sama maka akan dilakukan proses pemilihan sekali lagi.
4. Parent yang telah terpilih sebelumnya disimpan dalam array parent[][].
5. Proses mutasi pada parent yang telah dipilih dilakukan dengan menukar gen pada parent.

4.3.4 Seleksi

Seleksi yang digunakan pada penelitian ini adalah *elitism selection* yang akan memilih individu untuk generasi selanjutnya dengan mengurutkan nilai *fitness* yang dimiliki tiap individu. Pada Gambar 4.3 ditunjukkan seleksi individu dilakukan dengan memilih individu terbaik sesuai dengan jumlah pop size yang telah ditentukan.







Gambar 4.6 Diagram alir *elitism selection*

Langkah-langkah proses elitism selection yang ditunjukkan pada Gambar 4.5 adalah sebagai berikut:

1. Sistem menerima masukan kromosom berupa array `individu[][]` yang akan diseleksi.
2. Nilai *fitness* untuk individu diperoleh dari banyaknya *penalty* yang dilanggar.
3. Hasil nilai *fitness* akan diurutkan dari nilai besar ke kecil dengan membandingkan nilai *fitness* milik individu. Individu juga akan diurutkan berdasarkan *fitness* yang memiliki nilai terbesar.
4. Individu yang telah disorting disimpan dalam array `hasilSeleksi[][]`.

4.4 Perhitungan Manual

4.4.1 Inisialisasi Populasi Awal

1. Populasi mata pelajaran

Populasi awal mata pelajaran pada data perhitungan manual ini memiliki `popSize` sejumlah 3 yang direpresentasikan pada array berbentuk memanjang kesamping. Inisialisasi populasi awal mata pelajaran dapat dilihat pada Tabel 4.12.

Tabel 4.8 Populasi awal mata pelajaran

	X Bahasa														
	Hari 1		Hari 2		Hari 3		Hari 4		Hari 5		Hari 6		Hari 7		H8
P1	14	3	2	13	13	8	4	13	9	6	9	8	5	3	11
P2	14	14	7	6	9	5	5	4	6	6	8	1	8	11	7
P3	1	11	14	3	12	10	9	11	10	10	9	8	5	1	11
	XI Bahasa														
P1	29	26	16	27	28	21	23	16	18	27	29	28	18	22	
P2	23	20	16	21	18	27	23	26	23	26	17	24	16	27	
P3	25	16	26	22	21	22	25	16	29	26	28	18	28	25	
	XII Bahasa														
P1	36	32	30	36	30	32	42	30	31	37	32	33	36	43	
P2	40	31	31	40	42	34	33	32	42	31	34	34	40	43	
P3	39	33	42	42	36	34	41	32	32	30	34	31	34	42	
	X MIPA														
P1	47	53	55	46	44	51	56	58	55	47	58	55	54	58	48
P2	49	44	44	57	49	49	47	44	50	57	44	45	48	44	47
P3	52	57	57	44	54	47	55	48	51	55	56	50	46	46	54
	XI MIPA														
P1	69	63	61	66	71	59	70	69	60	71	70	67	65	67	
P2	65	64	67	70	62	60	71	59	62	69	68	72	68	70	
P3	72	63	65	71	69	60	66	66	63	64	72	71	65	59	
	XII MIPA														
P1	80	83	75	79	75	79	76	75	75	84	77	76	86	74	
P2	82	80	85	83	81	74	74	78	74	82	81	86	78	79	
P3	78	75	74	73	82	86	77	74	73	76	84	77	76	75	
	X IPS														
P1	92	97	94	98	89	92	95	98	89	97	101	99	93	92	99
P2	100	100	100	98	92	100	101	97	92	90	88	99	90	87	98
P3	87	96	89	91	99	101	98	95	90	91	97	98	91	101	100
	XI IPS														
P1	103	111	109	115	115	107	112	110	102	112	104	103	105	115	
P2	115	108	110	105	104	109	111	108	111	109	102	109	106	104	
P3	106	108	104	106	102	115	107	105	111	104	114	110	113	112	
	XII IPS														
P1	123	122	120	126	117	121	117	119	116	123	116	120	119	116	
P2	122	118	121	118	127	118	123	119	120	117	117	121	128	127	
P3	118	116	116	116	121	126	123	116	116	121	124	119	117	120	

2. Populasi pengawas

Populasi awal pengawas pada data perhitungan manual ini memiliki popSize sejumlah 2 yang direpresentasikan pada array berbentuk memanjang kesamping.

Populasi awal pengawas terbentuk setelah populasi mata pelajaran telah terbentuk. Inisialisasi populasi awal mata pelajaran dapat dilihat pada Tabel 4.13.

Tabel 4.9 Populasi awal pengawas

Sesi ke-1															
Hari ke- 1															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	78	91	167	164	63	154	230	141	34	28	53	60	175	76	219
P2	151	8	135	3	231	156	53	153	90	59	45	20	92	137	173
P3	4	50	221	161	47	109	62	75	224	17	173	160	152	20	177
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	222	71	3	26	181	145	107	64	200	116	49	161	86	127	214
P2	187	83	197	162	192	107	200	204	70	144	176	175	157	23	118
P3	14	109	88	32	209	206	68	13	29	63	81	72	83	80	89
Hari ke-2															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	42	200	98	98	17	13	64	171	176	179	54	150	100	217	61
P2	140	93	6	202	201	135	12	189	29	29	126	190	107	23	79
P3	16	27	70	180	178	122	185	101	81	231	149	227	38	218	193
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	205	227	208	96	185	97	208	229	62	186	33	142	163	99	131
P2	97	68	207	192	196	36	144	165	27	103	202	225	61	173	128
P3	134	73	213	227	82	170	194	209	230	56	2	19	160	182	156
Hari Ke-3															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	216	69	110	163	119	140	46	50	210	158	139	103	151	80	34
P2	168	199	59	95	95	84	158	40	128	172	175	21	147	131	73
P3	35	55	80	48	16	225	57	176	2	36	70	215	217	18	65
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	34	206	144	107	224	103	162	144	167	165	186	88	62	13	34
P2	72	133	37	5	231	154	189	11	179	73	98	166	132	17	72
P3	228	159	36	173	43	118	228	190	108	88	207	101	214	137	228
Hari ke-4															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	211	131	18	38	83	36	141	184	109	125	48	127	122	1	148
P2	140	8	39	109	202	160	39	170	214	132	111	111	59	200	195
P3	158	183	147	66	162	203	216	104	116	165	161	118	129	214	112
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	76	216	2	40	225	120	8	163	110	150	59	160	85	72	69
P2	101	135	202	30	189	93	16	174	113	13	167	172	142	12	96
P3	14	201	109	222	231	121	215	94	207	163	170	226	172	217	6

Hari Ke-5															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	80	93	76	120	184	73	114	189	223	104	9	97	113	8	201
P2	162	191	52	113	151	192	100	210	50	223	158	40	179	220	26
P3	37	30	223	70	30	122	181	109	204	15	213	40	130	106	214
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	127	176	87	36	70	134	105	75	163	132	32	34	47	78	56
P2	130	80	143	203	120	80	2	66	32	156	90	104	152	41	67
P3	33	197	61	157	51	32	97	212	27	12	30	118	142	137	216
Hari ke-6															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	98	163	112	113	195	80	54	16	90	58	227	176	219	6	98
P2	46	108	230	91	29	223	51	137	64	217	141	116	211	95	121
P3	61	223	166	23	66	146	226	9	140	10	212	176	178	89	165
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	41	201	212	146	130	150	102	22	69	127	227	177	166	50	112
P2	219	162	116	115	176	3	179	73	68	209	125	18	45	132	159
P3	3	146	173	224	167	3	103	196	226	113	78	102	45	179	74
Hari ke-7															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	178	182	192	50	61	102	217	170	214	61	122	85	40	59	183
P2	67	130	67	94	114	147	123	31	223	83	167	220	4	118	177
P3	10	119	64	200	183	43	134	162	148	69	76	16	221	214	209
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	223	107	65	207	154	35	63	199	113	89	23	83	125	76	141
P2	8	158	165	200	65	219	191	113	12	42	188	205	130	227	34
P3	192	188	112	196	122	194	212	122	29	194	217	184	102	115	16
Hari ke-8															
	X Bhs	X MIPA							X IPS						
P1	174	9	170	222	90	111	160	116	178	205					
P2	82	59	28	76	19	205	87	86	146	111					
P3	88	39	136	186	209	109	54	16	180	151					
Sesi 2															
Hari ke-1															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	187	174	89	8	88	119	69	154	88	217	183	198	175	118	59
P2	35	120	1	73	224	150	228	80	91	28	42	116	7	192	225
P3	5	37	20	128	64	133	64	57	41	226	160	151	104	74	32
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	1	145	90	184	114	102	119	76	149	131	188	194	127	72	177
P2	147	55	90	139	157	27	106	73	122	30	61	155	137	191	228

P3	175	101	21	101	35	227	37	228	197	30	134	98	112	15	55
Hari ke-2															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	60	171	184	139	56	120	141	217	112	80	80	143	41	198	132
P2	190	161	76	66	182	90	33	149	222	110	199	137	200	128	45
P3	197	96	169	19	181	11	24	38	160	132	38	65	130	122	190
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	129	5	202	220	63	134	173	224	4	155	7	210	104	160	12
P2	75	183	110	201	17	99	21	24	195	137	1	27	104	124	3
P3	141	54	102	116	223	108	105	229	53	202	84	60	193	82	16
Hari ke-3															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	142	216	85	19	87	108	22	83	176	230	178	11	230	228	125
P2	165	136	20	74	141	71	102	168	89	24	195	80	184	117	203
P3	227	224	93	76	143	218	116	211	191	218	55	4	205	206	39
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	204	39	23	28	230	127	213	99	193	169	167	152	9	89	17
P2	141	153	162	97	194	63	9	80	219	142	1	104	29	98	161
P3	171	76	150	228	25	213	74	11	15	39	85	62	97	228	51
Hari ke-4															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	178	68	104	204	29	30	104	142	125	108	164	126	35	80	212
P2	224	228	88	148	200	152	37	36	6	79	55	226	110	66	100
P3	196	102	203	128	6	90	98	128	167	14	89	165	159	13	54
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	2	133	193	198	116	152	208	82	38	182	76	131	5	226	14
P2	150	211	208	143	58	225	192	10	206	118	196	121	34	195	9
P3	13	141	155	137	34	204	40	128	12	110	161	113	199	15	87
Hari ke-5															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	22	70	96	11	88	30	146	98	50	54	145	28	131	64	118
P2	17	125	209	219	145	180	104	27	22	173	199	209	202	108	14
P3	57	15	169	171	52	153	73	105	93	29	93	25	173	29	197
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	146	110	174	129	68	101	149	136	73	102	218	223	178	33	106
P2	34	30	227	74	44	63	11	172	171	129	118	8	227	218	150
P3	140	31	218	129	173	138	25	126	96	178	100	120	61	158	13
Hari ke-6															
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	58	130	19	158	118	34	30	17	157	64	44	77	136	13	72
P2	98	110	5	191	36	199	220	226	196	197	138	165	161	211	136

P3	15	175	44	206	12	224	69	17	135	21	203	122	180	203	130
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	56	205	39	10	205	11	229	119	221	162	143	217	199	183	41
P2	131	130	43	116	89	199	116	114	133	155	66	39	28	211	145
P3	95	209	26	17	222	105	93	68	218	129	43	97	152	26	15
	Hari ke-7														
	X Bhs	XI Bhs	XII Bhs	X MIPA						XI MIPA					
P1	128	159	189	185	132	110	4	207	226	50	84	114	96	29	135
P2	55	149	147	157	108	11	221	217	209	208	45	191	98	139	143
P3	77	65	133	63	53	166	49	41	74	111	58	69	220	104	36
	XII MIPA						X IPS			XI IPS			XII IPS		
P1	140	131	115	223	162	145	219	176	93	61	203	216	30	166	142
P2	198	126	3	210	80	38	174	84	10	39	44	209	149	62	218
P3	98	205	130	131	173	215	20	191	148	125	85	162	192	75	185

4.4.2 Crossover

Pada perhitungan manual ini *crossover rate* ditentukan sebesar 0.6, sehingga *offspring* yang dihasilkan pada proses *crossover* adalah $3 \times 0.6 = 1.8 \approx 2$ *offspring*. Pemilihan induk *crossover* dilakukan secara acak.

1. Crossover Mata Pelajaran

Parent yang dipilih secara acak untuk dilakukan *crossover* adalah *parent* 1 dan *parent* 2. Proses *crossover* mata pelajaran merupakan *multi-point crossover* dan ditunjukkan pada Tabel 4.13 berikut.

Tabel 4.10 Proses *multi-point crossover*

	X Bahasa														
P1	14	14	7	6	9	5	5	4	6	6	8	1	8	11	7
P2	1	11	14	3	12	10	9	11	10	10	9	8	5	1	11
C1	14	14	7	6	9	5	5	4	6	6	8	1	8	1	11
C2	1	11	14	3	12	10	9	11	10	10	9	8	5	11	7
	XI Bahasa														
P1	23	20	16	21	18	27	23	26	23	26	17	24	16	27	
P2	25	16	26	22	21	22	25	16	29	26	28	18	28	25	
C1	23	20	16	21	18	27	23	26	23	26	17	18	28	25	
C2	25	16	26	22	21	22	25	16	29	26	28	24	16	27	
	XII Bahasa														
P1	40	31	31	40	42	34	33	32	42	31	34	34	40	43	
P2	39	33	42	42	36	34	41	32	32	30	34	31	34	42	
C1	40	31	31	40	42	34	33	32	42	30	34	31	34	42	
C2	39	33	42	42	36	34	41	32	32	31	34	34	40	43	
	X MIPA														

P1	49	44	44	57	49	49	47	44	50	57	44	45	48	44	47
P2	52	57	57	44	54	47	55	48	51	55	56	50	46	46	54
C1	49	44	44	57	54	47	55	48	51	55	56	50	46	46	54
C2	52	57	57	44	49	49	47	44	50	57	44	45	48	44	47
XI MIPA															
P1	65	64	67	70	62	60	71	59	62	69	68	72	68	70	
P2	72	63	65	71	69	60	66	66	63	64	72	71	65	59	
C1	65	64	67	70	69	60	66	66	63	64	72	71	65	59	
C2	72	63	65	71	62	60	71	59	62	69	68	72	68	70	
XII MIPA															
P1	82	80	85	83	81	74	74	78	74	82	81	86	78	79	
P2	78	75	74	73	82	86	77	74	73	76	84	77	76	75	
C1	82	80	85	83	81	86	77	74	73	76	84	77	76	75	
C2	78	75	74	73	82	74	74	78	74	82	81	86	78	79	
X IPS															
P1	100	100	100	98	92	100	101	97	92	90	88	99	90	87	98
P2	87	96	89	91	99	101	98	95	90	91	97	98	91	101	100
C1	100	100	100	98	92	101	98	95	90	91	97	98	91	101	100
C2	87	96	89	91	99	100	101	97	92	90	88	99	90	87	98
XI IPS															
P1	115	108	110	105	104	109	111	108	111	109	102	109	115	108	
P2	106	108	104	106	102	115	107	105	111	104	114	110	106	108	
C1	115	108	110	105	104	109	111	108	111	104	114	110	115	108	
C2	106	108	104	106	102	115	107	105	111	109	102	109	106	108	
XII IPS															
P1	106	104	122	118	121	118	127	118	123	119	120	117	117	121	
P2	113	112	118	116	116	116	121	126	123	116	116	121	124	119	
C1	113	112	122	118	121	118	121	126	123	116	116	121	124	119	
C2	106	104	118	116	116	116	127	118	123	119	120	117	117	121	

2. Crossover Pengawas

Parent yang dipilih secara acak untuk dilakukan *crossover* adalah *parent* 2 dan *parent* 1. Proses *crossover* pengawas merupakan *one-cut-point crossover* dan ditunjukkan pada Tabel 4.14 berikut.

Tabel 4.11 Proses *one-cut-point crossover*

P2	151	8	135	3	231	156	53	153	90	59	45	20	92	137	173
	187	83	197	162	192	107	200	204	70	144	176	175	157	23	118
	140	93	6	202	201	135	12	189	29	29	126	190	107	23	79
	97	68	207	192	196	36	144	165	27	103	202	225	61	173	128
	168	199	59	95	95	84	158	40	128	172	175	21	147	131	73
	72	133	37	5	231	154	189	11	179	73	98	166	132	17	140
	8	39	109	202	160	39	170	214	132	111	111	59	200	195	94

	101	135	202	30	189	93	16	174	113	13	167	172	142	12	96
	162	191	52	113	151	192	100	210	50	223	158	40	179	220	26
	130	80	143	203	120	80	2	66	32	156	90	104	152	41	67
	46	108	230	91	29	223	51	137	64	217	141	116	211	95	121
	219	162	116	115	176	3	179	73	68	209	125	18	45	132	159
	67	130	67	94	114	147	123	31	223	83	167	220	4	118	177
	8	158	165	200	65	219	191	113	12	42	188	205	130	227	34
	82	59	28	76	19	205	87	86	146	111	35	120	1	73	224
	150	228	80	91	28	42	116	7	192	225	147	55	90	139	157
	27	106	73	122	30	61	155	137	191	228	190	161	76	66	182
	90	33	149	222	110	199	137	200	128	45	75	183	110	201	17
	99	21	24	195	137	1	27	104	124	3	165	136	20	74	141
	71	102	168	89	24	195	80	184	117	203	141	153	162	97	194
	63	9	80	219	142	1	104	29	98	161	224	228	88	148	200
	152	37	36	6	79	55	226	110	66	100	150	211	208	143	58
	225	192	10	206	118	196	121	34	195	9	17	125	209	219	145
	180	104	27	22	173	199	209	202	108	14	34	30	227	74	44
	118	8	227	218	150	98	110	5	191	36	199	220	226	196	197
	138	165	161	211	136	131	130	43	116	89	199	116	114	133	155
	66	39	28	211	145	55	149	147	157	108	11	221	217	209	208
	45	191	98	139	143	198	126	3	210	80	38	174	84	10	39
	44	209	149	62	218										
P1	78	91	167	164	63	154	230	141	34	28	53	60	175	76	219
	222	71	3	26	181	145	107	64	200	116	49	161	86	127	214
	42	200	98	98	17	13	64	171	176	179	54	150	100	217	61
	205	227	208	96	185	97	208	229	62	186	33	142	163	99	131
	216	69	110	163	119	140	46	50	210	158	139	103	151	80	34
	34	206	144	107	224	103	162	144	167	165	186	88	62	13	211
	131	18	38	83	36	141	184	109	125	48	127	122	1	148	124
	76	216	2	40	225	120	8	163	110	150	59	160	85	72	69
	80	93	76	120	184	73	114	189	223	104	9	97	113	8	201
	12	8	87	36	70	134	105	75	163	132	32	34	47	78	56
	98	163	112	113	195	80	54	16	90	58	227	176	219	6	98
	41	201	212	146	130	150	102	22	69	127	227	177	166	50	112
	178	182	192	50	61	102	217	170	214	61	122	85	40	59	183
	223	107	65	207	154	35	63	199	113	89	23	83	125	76	141
	174	9	170	222	90	111	160	116	178	205	187	174	89	8	88
	119	69	154	88	217	183	198	175	118	59	1	145	90	184	114
	102	119	76	149	131	188	194	127	72	177	60	171	184	139	56
	141	217	112	80	80	143	41	198	132	129	5	202	220	63	134
	7	210	104	160	12	142	216	85	19	87	108	22	83	176	230
	178	11	230	228	125	204	39	23	28	230	127	213	99	193	169
	167	152	9	89	17	178	68	104	204	29	30	104	142	125	108

	164	126	35	80	212	2	133	193	198	116	152	208	82	38	182
	76	131	5	226	14	22	70	96	11	88	30	146	98	50	54
	145	28	131	64	118	146	110	174	129	68	101	149	136	73	102
	218	223	178	33	106	58	130	19	158	118	34	30	17	157	64
	44	77	136	13	72	56	205	39	10	205	11	229	119	221	162
	143	217	199	183	41	128	159	189	185	132	110	4	207	226	50
	84	114	96	29	135	140	131	115	223	162	145	219	176	93	61
	203	216	30	166	142										
C1	151	8	135	3	231	156	53	153	90	59	45	20	92	137	173
	187	83	197	162	192	107	200	204	70	144	176	175	157	23	118
	140	93	6	202	201	135	12	189	29	29	126	190	107	23	79
	205	227	208	96	185	97	208	229	62	186	33	142	163	99	131
	216	69	110	163	119	140	46	50	210	158	139	103	151	80	34
	34	206	144	107	224	103	162	144	167	165	186	88	62	13	211
	131	18	38	83	36	141	184	109	125	48	127	122	1	148	124
	76	216	2	40	225	120	8	163	110	150	59	160	85	72	69
	80	93	76	120	184	73	114	189	223	104	9	97	113	8	201
	12	8	87	36	70	134	105	75	163	132	32	34	47	78	56
	98	163	112	113	195	80	54	16	90	58	227	176	219	6	98
	41	201	212	146	130	150	102	22	69	127	227	177	166	50	112
	178	182	192	50	61	102	217	170	214	61	122	85	40	59	183
	223	107	65	207	154	35	63	199	113	89	23	83	125	76	141
	174	9	170	222	90	111	160	116	178	205	187	174	89	8	88
	119	69	154	88	217	183	198	175	118	59	1	145	90	184	114
	102	119	76	149	131	188	194	127	72	177	60	171	184	139	56
	141	217	112	80	80	143	41	198	132	129	5	202	220	63	134
	7	210	104	160	12	142	216	85	19	87	108	22	83	176	230
	178	11	230	228	125	204	39	23	28	230	127	213	99	193	169
	167	152	9	89	17	178	68	104	204	29	30	104	142	125	108
	164	126	35	80	212	2	133	193	198	116	152	208	82	38	182
	76	131	5	226	14	22	70	96	11	88	30	146	98	50	54
	145	28	131	64	118	146	110	174	129	68	101	149	136	73	102
	218	223	178	33	106	58	130	19	158	118	34	30	17	157	64
	44	77	136	13	72	56	205	39	10	205	11	229	119	221	162
	143	217	199	183	41	128	159	189	185	132	110	4	207	226	50
	84	114	96	29	135	140	131	115	223	162	145	219	176	93	61
	203	216	30	166	142										
C2	78	91	167	164	63	154	230	141	34	28	53	60	175	76	219
	222	71	3	26	181	145	107	64	200	116	49	161	86	127	214
	42	200	98	98	17	13	64	171	176	179	54	150	100	217	61
	97	68	207	192	196	36	144	165	27	103	202	225	61	173	128
	168	199	59	95	95	84	158	40	128	172	175	21	147	131	73
	72	133	37	5	231	154	189	11	179	73	98	166	132	17	140

	8	39	109	202	160	39	170	214	132	111	111	59	200	195	94
	101	135	202	30	189	93	16	174	113	13	167	172	142	12	96
	162	191	52	113	151	192	100	210	50	223	158	40	179	220	26
	130	80	143	203	120	80	2	66	32	156	90	104	152	41	67
	46	108	230	91	29	223	51	137	64	217	141	116	211	95	121
	219	162	116	115	176	3	179	73	68	209	125	18	45	132	159
	67	130	67	94	114	147	123	31	223	83	167	220	4	118	177
	8	158	165	200	65	219	191	113	12	42	188	205	130	227	34
	82	59	28	76	19	205	87	86	146	111	35	120	1	73	224
	150	228	80	91	28	42	116	7	192	225	147	55	90	139	157
	27	106	73	122	30	61	155	137	191	228	190	161	76	66	182
	90	33	149	222	110	199	137	200	128	45	75	183	110	201	17
	99	21	24	195	137	1	27	104	124	3	165	136	20	74	141
	71	102	168	89	24	195	80	184	117	203	141	153	162	97	194
	63	9	80	219	142	1	104	29	98	161	224	228	88	148	200
	152	37	36	6	79	55	226	110	66	100	150	211	208	143	58
	225	192	10	206	118	196	121	34	195	9	17	125	209	219	145
	180	104	27	22	173	199	209	202	108	14	34	30	227	74	44
	118	8	227	218	150	98	110	5	191	36	199	220	226	196	197
	138	165	161	211	136	131	130	43	116	89	199	116	114	133	155
	66	39	28	211	145	55	149	147	157	108	11	221	217	209	208
	45	191	98	139	143	198	126	3	210	80	38	174	84	10	39
	44	209	149	62	218										

4.4.3 Mutasi

1. Mutasi mata pelajaran

Pada perhitungan manual ini *mutation rate* ditentukan sebesar 0.4, sehingga *offspring* yang dihasilkan pada proses mutasi adalah $3 \times 0.4 = 1.2 \approx 1$ *offspring*. Pemilihan induk mutasi dilakukan secara acak. Parent yang dipilih secara acak untuk dilakukan mutasi adalah parent 1. Proses *random exchange mutation* ditunjukkan pada Tabel 4.16 berikut.

Tabel 4.12 Proses *random exchange mutation* mata pelajaran

	X Bahasa														
P1	14	3	2	13	13	8	4	13	9	6	9	8	5	3	11
C3	14	3	2	13	13	8	4	13	9	6	9	8	5	3	11
	XI Bahasa														
P1	29	26	16	27	28	21	23	16	18	27	29	28	18	22	
C3	29	28	16	27	28	21	23	16	18	27	29	26	18	22	
	XII Bahasa														
P1	36	32	30	36	30	32	42	30	31	37	32	33	36	43	
C3	36	32	30	31	30	32	42	30	36	37	32	33	36	43	

	X MIPA														
P1	47	53	55	46	44	51	56	58	55	47	58	55	54	58	48
C3	47	58	55	46	44	51	56	58	55	47	53	55	54	58	48
	XI MIPA														
P1	69	63	61	66	71	59	70	69	60	71	70	67	65	67	
C3	69	63	61	66	71	59	70	69	60	71	70	67	65	67	
	XII MIPA														
P1	80	83	75	79	75	79	76	75	75	84	77	76	86	74	
C3	80	79	75	83	75	79	76	75	75	84	77	76	86	74	
	X IPS														
P1	92	97	94	98	89	92	95	98	89	97	101	99	93	92	99
C3	92	97	94	98	98	92	95	89	89	97	101	99	93	92	99
	XI IPS														
P1	103	111	109	115	115	107	112	110	102	112	104	103	105	115	
C3	103	111	115	115	109	107	112	110	102	112	104	103	105	115	
	XII IPS														
P1	123	122	120	126	117	121	117	119	116	123	116	120	119	116	
C3	123	120	120	126	117	121	117	119	116	123	116	122	119	116	

2. Mutasi pengawas

Pada perhitungan manual ini *mutation rate* ditentukan sebesar 0.4, sehingga *offspring* yang dihasilkan pada proses mutasi adalah $3 \times 0.4 = 1.2 \approx 1$ *offspring*. Pemilihan induk mutasi dilakukan secara acak. Parent yang dipilih secara acak untuk dilakukan mutasi adalah parent 3. Proses *random exchange mutation* ditunjukkan pada Tabel 4.17 berikut.

Tabel 4.13 Proses *random exchange mutation* pengawas

P2	151	8	135	3	231	156	53	153	90	59	45	20	92	137	173
	187	83	197	162	192	107	200	204	70	144	176	175	157	23	118
	140	93	6	202	201	135	12	189	29	29	126	190	107	23	79
	97	68	207	192	196	36	144	165	27	103	202	225	61	173	128
	168	199	59	95	95	84	158	40	128	172	175	21	147	131	73
	72	133	37	5	231	154	189	11	179	73	98	166	132	17	140
	8	39	109	202	160	39	170	214	132	111	111	59	200	195	94
	101	135	202	30	189	93	16	174	113	13	167	172	142	12	96
	162	191	52	113	151	192	100	210	50	223	151	40	179	220	26
	130	80	143	203	120	80	2	66	32	156	90	104	152	41	67
	46	108	230	91	29	223	51	137	64	217	141	116	211	95	121
	219	162	116	115	176	3	179	73	68	209	125	18	45	132	159
	67	130	67	94	114	147	123	31	223	83	167	220	4	118	177
	8	158	165	200	65	219	191	113	12	42	188	205	130	227	34
	82	59	28	76	19	205	87	86	146	111	35	120	1	73	224
	150	228	80	91	28	42	116	7	192	225	147	55	90	139	157

	27	106	73	122	30	61	155	137	191	228	190	161	76	66	182
	90	33	149	222	110	199	137	200	128	45	75	183	110	201	17
	99	21	24	195	137	1	27	104	124	3	165	136	20	74	141
	71	102	168	89	24	195	80	184	117	203	141	153	162	97	194
	63	9	80	219	142	1	104	29	98	161	224	228	88	148	200
	152	37	36	6	79	55	226	110	66	100	150	211	208	143	58
	225	192	10	206	118	196	121	34	195	9	17	125	209	219	145
	180	104	27	22	173	199	209	202	108	14	34	30	227	74	44
	118	8	227	218	150	98	110	5	191	36	199	220	226	196	197
	138	165	161	211	136	131	130	43	116	89	199	116	114	133	155
	66	39	28	211	145	55	149	147	157	108	11	221	217	209	208
	45	191	98	139	143	198	126	3	210	80	38	174	84	10	39
	44	209	149	62	218										
C3	151	8	135	3	231	156	53	153	90	59	45	20	92	137	173
	187	83	197	162	192	107	200	204	70	144	176	175	157	23	118
	140	93	6	202	201	135	12	189	29	29	126	190	107	23	79
	97	68	207	192	196	36	144	165	27	103	202	225	61	173	128
	168	199	59	95	95	84	151	40	128	172	175	21	147	131	73
	72	133	37	5	231	154	189	11	179	73	98	166	132	17	140
	8	39	109	202	160	39	170	214	132	111	111	59	200	195	94
	101	135	202	30	189	93	16	174	113	13	167	172	142	12	96
	162	191	52	113	151	192	100	210	50	223	158	40	179	220	26
	130	80	143	203	120	80	2	66	32	156	90	104	152	41	67
	46	108	230	91	29	223	51	137	64	217	141	116	211	95	121
	219	162	116	115	176	3	179	73	68	209	125	18	45	132	159
	67	130	67	94	114	147	123	31	223	83	167	220	4	118	177
	8	158	165	200	65	219	191	113	12	42	188	205	130	227	34
	82	59	28	76	19	205	87	86	146	111	35	120	1	73	224
	150	228	80	91	28	42	116	7	192	225	147	55	90	139	157
	27	106	73	122	30	61	155	137	191	228	190	161	76	66	182
	90	33	149	222	110	199	137	200	128	45	75	183	110	201	17
	99	21	24	195	137	1	27	104	124	3	165	136	20	74	141
	71	102	168	89	24	195	80	184	117	203	141	153	162	97	194
	63	9	80	219	142	1	104	29	98	161	224	228	88	148	200
	152	37	36	6	79	55	226	110	66	100	150	211	208	143	58
	225	192	10	206	118	196	121	34	195	9	17	125	209	219	145
	180	104	27	22	173	199	209	202	108	14	34	30	227	74	44
	118	8	227	218	150	98	110	5	191	36	199	220	226	196	197
	138	165	161	211	136	131	130	43	116	89	199	116	114	133	155
	66	39	28	211	145	55	149	147	157	108	11	221	217	209	208
	45	191	98	139	143	198	126	3	210	80	38	174	84	10	39
	44	209	149	62	218										

4.4.4 Seleksi

Pada proses seleksi dengan elitism selection akan memilih individu terbaik dengan nilai *fitness* terbesar yang telah dihitung sebelumnya dengan rumus 4.1. *Fitness* diperoleh dengan membagi 1 dengan bilangan 1 yang telah ditambahkan dengan jumlah *penalty* dari masing-masing individu.

$$fitness = \frac{1}{1 + \sum \text{penalty}} \quad 4.1$$

Individu parent dan *offspring* hasil reproduksi akan diurutkan berdasarkan nilai *fitness* yang dimiliki. Individu-individu tersebut akan dipilih berdasarkan jumlah popSize yang telah ditentukan. Tabel 4.17 dan Tabel 4.18 menunjukkan hasil perhitungan *fitness* dan hasil evaluasi untuk individu mata pelajaran. Sedangkan pada Tabel 4.19 dan Tabel 4.20 menunjukkan hasil perhitungan *fitness* dan hasil evaluasi untuk individu pengawas.

Tabel 4.14 Hasil perhitungan *fitness* individu mata pelajaran

Individu	<i>Fitness</i>
P1	0.053384583
P2	0.050676532
P3	0.057796786
C1	0.057323015
C2	0.056398398
C3	0.053795255

Tabel 4.15 Hasil seleksi individu mata pelajaran

No	Individu	<i>Fitness</i>
1	P3	0.057796786
2	C1	0.057323015
3	C2	0.056398398

Tabel 4.16 Hasil perhitungan *fitness* individu pengawas

Individu	<i>Fitness</i>
P1	0.00109589
P2	0.001133787
P3	0.001109878
C1	0.001113586
C2	0.001121076
C3	0.001139601

Tabel 4.17 Hasil seleksi individu mata pengawas

No	Individu	<i>Fitness</i>
1	P1	0.00109589
2	P3	0.001109878

3	C1	0.001113586
---	----	-------------

4.5 Perancangan Pengujian

4.5.1 Perancangan Pengujian Ukuran Populasi

Pengujian jumlah populasi dilakukan guna mendapatkan ukuran populasi yang optimal. Tabel 4.23 menunjukkan rancangan pengujian ukuran populasi dengan jumlah populasi dalam rentang 10 hingga 200. Sedangkan nilai *crossover rate* (Cr) dan *mutation rate* (Mr) yang digunakan merupakan kombinasi Cr dan Mr yang menghasilkan nilai *fitness* terbaik pada pengujian nilai Cr dan Mr. Jumlah generasi yang digunakan pada pengujian ini senilai jumlah generasi yang digunakan pada pengujian Cr dan Mr sebelumnya.

Tabel 4.18 Perancangan pengujian ukuran populasi

Jumlah Populasi	Percobaan ke-				Rata-rata
	1	2	...	5	
10					
20					
...					
200					

4.5.2 Perancangan Pengujian Kombinasi Crossover Rate (Cr) dan Mutation Rate (Mr)

Pengujian nilai *crossover rate* (cr) dan *mutation rate* (mr) dilakukan guna mengetahui nilai kombinasi *crossover rate* dan *mutation rate* yang paling optimal. Pada pengujian ini ukuran populasi yang digunakan senilai 200, sedangkan untuk jumlah generasi menggunakan 500 generasi. Kombinasi Cr dan Mr yang digunakan untuk pengujian ditampilkan pada Tabel 4.22.

Tabel 4.19 Perancangan pengujian kombinasi Cr dan Mr

Kombina si		Percobaan ke-										Rata - Rata
		1		2		3		..		5		
Cr	Mr	waktu	fitness	Waktu	fitness	waktu	fitness	waktu	fitness	waktu	fitness	
0.9	0.1											
0.8	0.2											
0.7	0.3											
0.6	0.4											



0.5	0.5											
0.4	0.6											
0.3	0.7											
0.2	0.8											
0.1	0.9											



BAB 5 IMPLEMENTASI

5.1 Lingkungan Sistem

5.1.1 Lingkungan Perangkat Keras

Sistem yang dibangun dalam penelitian ini menggunakan sebuah komputer dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Lingkungan perangkat keras

Komponen	Spesifikasi
Processor	Intel® Core™ i3-4005U CPU @ 1.70GHz 1.70GHz
Memori (RAM)	6.00 GB
Kartu Grafis	Intel® HD Graphics Family
Hardisk	500 GB

5.1.2 Lingkungan Perangkat Lunak

Sistem yang dibangun dalam penelitian ini menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Lingkungan perangkat lunak

Sistem Operasi	Windows 8.1 Pro
Bahasa Pemrograman	Java
Tools Pemrograman	Netbeans IDE 8.0.2

5.2 Implementasi Algoritme

5.2.1 Implementasi Inisialisasi Populasi Awal

1. Populasi Awal Mata Pelajaran

Populasi awal mata pelajaran tersusun atas kromosom-kromosom yang memiliki panjang 129. Populasi tersebut dibangkitkan dari bilangan random 1 sampai 129 seperti yang ditunjukkan pada Tabel 5.3.

Tabel 5.3 Implementasi kode program populasi awal mata pelajaran

1	int[][] individu = new int[popSize][jumlahMapel];
2	int chromLength = individu[0].length;
3	for (int i = 0; i < popSize; i++) {
4	for (int j = 0; j < individu[0].length; j++) {
5	if (j <= 14) {
6	individu[i][j]
7	random.nextInt(batasMaxBahasa10 - batasMinBahasa10 + 1) +
8	batasMinBahasa10;
9	} else if ((j > 14) && (j <= 28)) {
10	

11	individu[i][j]	=
12	random.nextInt(batasMaxBahasa11 - batasMinBahasa11 + 1) +	
13	batasMinBahasa11;	
14	} else if ((j > 28) && (j <= 42)) {	
15	individu[i][j]	=
16	random.nextInt(batasMaxBahasa12 - batasMinBahasa12 + 1) +	
17	batasMinBahasa12;	
18	} else if ((j > 42) && (j <= 57)) {	
19	individu[i][j] = random.nextInt(batasMaxMipa10	
20	- batasMinMipa10 + 1) + batasMinMipa10;	
21	} else if ((j > 57) && (j <= 71)) {	
22	individu[i][j] = random.nextInt(batasMaxMipa11	
23	- batasMinMipa11 + 1) + batasMinMipa11;	
24	} else if ((j > 71 && (j <= 85))) {	
25	individu[i][j] = random.nextInt(batasMaxMipa12	
26	- batasMinMipa12 + 1) + batasMinMipa12;	
27	} else if ((j > 85) && (j <= 100)) {	
28	individu[i][j] = random.nextInt(batasMaxIps10 -	
29	batasMinIps10 + 1) + batasMinIps10;	
30	} else if ((j > 100) && (j <= 114)) {	
31	individu[i][j] = random.nextInt(batasMaxIps11 -	
32	batasMinIps11 + 1) + batasMinIps11;	
33	} else if (j > 114) {	
34	individu[i][j] = random.nextInt(batasMaxIps12 -	
35	batasMinIps12 + 1) + batasMinIps12;	
36	}	
37	}	

2. Populasi Awal Pengawas

Populasi awal pengawas atas kromosom-kromosom yang memiliki panjang 430. Populasi ini dibangkitkan dari bilangan random 1 sampai 66 sejumlah pengawas yang ada seperti yang ditunjukkan pada Tabel 5.3.

Tabel 5.4 Implementasi kode program inisialisasi populasi awal pengawas

1	int batasMin = 1;
2	int batasMax = data.data.kombinasiPengawas - 1;
3	System.out.println("\nIndividuPengawas");
4	for (int i = 0; i < indvPengawas.length; i++) {
5	System.out.print("P" + (i + 1) + ": ");
6	for (int j = 0; j < indvPengawas[0].length; j++) {
7	indvPengawas[i][j] = random.nextInt(batasMax -
8	batasMin + 1) + batasMin;
9	System.out.print(indvPengawas[i][j] + "\t");
10	}
11	System.out.println("");
12	}

5.2.2 Implementasi Crossover

1. Multi-cut-point Crossover

Populasi awal mata pelajaran akan dilakukan proses pindah silang dengan *multi-point crossover* ini. Proses ini akan menghasilkan 1 *offspring* dari *crossover*. Implementasi metode *multi-point crossover* pada penelitian ini ditunjukkan pada Tabel 5.5.

Tabel 5.5 Implementasi kode program *multi-point crossover*

1	public int[][] MultiCrossover(int[][] individu) {			
2	Random random = new Random();			
3	int popSize = individu.length;			
4	int chromLength = individu[0].length;			
5	int[][] offspringCrossover = new int[2][chromLength];			
6	int batasMinParent, batasMaxParent;			
7	int[][] parent1 = new int[1][chromLength];			
8	int[][] parent2 = new int[1][chromLength];			
9	batasMinParent = 1;			
10	batasMaxParent = popSize - 1;			
11	int indexp1 = 0, indexp2 = 0;			
12	boolean kondisi = true;			
13	do {			
14	indexp1 = random.nextInt(batasMaxParent	-		
15	batasMinParent + 1) + batasMinParent;			
16	indexp2 = random.nextInt(batasMaxParent	-		
17	batasMinParent + 1) + batasMinParent;			
18	if (indexp1 != indexp2) {			
19	kondisi = false;			
20	}			
21	} while (kondisi);			
22	int[] posisi = new int[banyakKelas];			
23	posisi[0] = random.nextInt(batasMaxBahasa10	-	1	-
24	batasMinBahasa10 + 1) + batasMinBahasa10;			
25	posisi[1] = random.nextInt(batasMaxBahasa11	-	1	-
26	batasMinBahasa11 + 1) + batasMinBahasa11;			
27	posisi[2] = random.nextInt(batasMaxBahasa12	-	1	-
28	batasMinBahasa12 + 1) + batasMinBahasa12;			
29	posisi[3] = random.nextInt(batasMaxMipa10	-	1	-
30	batasMinMipa10 + 1) + batasMinMipa10;			
31	posisi[4] = random.nextInt(batasMaxMipa11	-	1	-
32	batasMinMipa11 + 1) + batasMinMipa11;			
33	posisi[5] = random.nextInt(batasMaxMipa12	-	1	-
34	batasMinMipa12 + 1) + batasMinMipa12;			
35	posisi[6] = random.nextInt(batasMaxIps10	-	1	-
36	batasMinIps10 + 1) + batasMinIps10;			
37	posisi[7] = random.nextInt(batasMaxIps11	-	1	-
38	batasMinIps11 + 1) + batasMinIps11;			
39	posisi[8] = random.nextInt(batasMaxIps12	-	1	-
40	batasMinIps12 + 1) + batasMinIps12;			
41	for (int i = 0; i < chromLength; i++) {			
42	parent1[0][i] = individu[indexp1][i];			
43	offspringCrossover[0][i] = individu[indexp1][i];			
44	parent2[0][i] = individu[indexp2][i];			
45	offspringCrossover[1][i] = individu[indexp2][i];			
46	}			
47	for (int k = posisi[0] - 1; k < batasMaxBahasa10; k++) {			
48	offspringCrossover[0][k] = parent2[0][k];			
49	offspringCrossover[1][k] = parent1[0][k];			
50	}			
51	for (int k = posisi[1] - 1; k < batasMaxBahasa11; k++) {			
52	offspringCrossover[0][k] = parent2[0][k];			
53	offspringCrossover[1][k] = parent1[0][k];			
54	}			
55	for (int k = posisi[2] - 1; k < batasMaxBahasa12; k++) {			
56	offspringCrossover[0][k] = parent2[0][k];			
57	offspringCrossover[1][k] = parent1[0][k];			
58	}			
59	for (int k = posisi[3] - 1; k < batasMaxMipa10; k++) {			
60	offspringCrossover[0][k] = parent2[0][k];			
61	offspringCrossover[1][k] = parent1[0][k];			
62	}			
63	for (int k = posisi[4] - 1; k < batasMaxMipa11; k++) {			



64	offspringCrossover[0][k] = parent2[0][k];
65	offspringCrossover[1][k] = parent1[0][k];
66	}
67	for (int k = posisi[5] - 1; k < batasMaxMipal2; k++) {
68	offspringCrossover[0][k] = parent2[0][k];
69	offspringCrossover[1][k] = parent1[0][k];
70	}
71	for (int k = posisi[6] - 1; k < batasMaxIps10; k++) {
72	offspringCrossover[0][k] = parent2[0][k];
73	offspringCrossover[1][k] = parent1[0][k];
74	}
75	for (int k = posisi[7] - 1; k < batasMaxIps11; k++) {
76	offspringCrossover[0][k] = parent2[0][k];
77	offspringCrossover[1][k] = parent1[0][k];
78	}
79	for (int k = posisi[8] - 1; k < batasMaxIps12; k++) {
80	offspringCrossover[0][k] = parent2[0][k];
81	offspringCrossover[1][k] = parent1[0][k];
82	}
83	return offspringCrossover;
84	}

2. One-cut-point Crossover

Populasi awal pengawas akan dilakukan proses pindah silang dengan metode *one-cut-point crossover* ini. Proses ini akan menghasilkan 1 *offspring* dari *crossover*. Implementasi metode *one-cut-point crossover* pada penelitian ini ditunjukkan pada Tabel 5.6.

Tabel 5.6 Implementasi kode program *one-cut-point crossover*

1	public int[][] OneCutPointCrossover(int[][] individu) {
2	Random random = new Random();
3	Data data = new Data();
4	int popSize = individu.length;
5	int chromLength = individu[0].length;
6	int[][] offspringCrossover = new int[2][chromLength];
7	int batasMinParent, batasMaxParent;
8	int[][] parent1 = new int[1][chromLength];
9	int[][] parent2 = new int[1][chromLength];
10	batasMinParent = 1;
11	batasMaxParent = popSize - 1;
12	int indexp1 = 0, indexp2 = 0;
13	boolean kondisi = true;
14	do {
15	indexp1 = random.nextInt(batasMaxParent -
16	batasMinParent + 1) + batasMinParent;
17	indexp2 = random.nextInt(batasMaxParent -
18	batasMinParent + 1) + batasMinParent;
19	if (indexp1 != indexp2) {
20	kondisi = false;
21	}
22	} while (kondisi);
23	int batasMax = data.jumlahPengawas - 1;
24	int batasMin = 1;
25	int pointCrossover = random.nextInt(batasMax - batasMin +
26	1) + batasMin;
27	for (int i = 0; i < chromLength; i++) {
28	parent1[0][i] = individu[indexp1 - 1][i];
29	offspringCrossover[0][i] = individu[indexp1 - 1][i];
30	parent2[0][i] = individu[indexp2 - 1][i];
31	offspringCrossover[1][i] = individu[indexp2 - 1][i];
32	}

33	for (int k = pointCrossover - 1; k < chromLength; k++) {
34	offspringCrossover[0][k] = parent2[0][k];
35	offspringCrossover[1][k] = parent1[0][k];
36	}
37	return offspringCrossover;
38	}

5.2.3 Implementasi Mutasi

Populasi awal kromosom mata pelajaran dan pengawas akan dilakukan proses mutasi dengan metode *random exchange mutation* ini. Proses ini akan menghasilkan 1 *offspring* dari mutasi. Implementasi metode *random exchange mutation* pada penelitian ini ditunjukkan pada Tabel 5.5.

Tabel 5.7 Implementasi kode program *random exchange mutation*

1	public int[][] RandomExchangeMutation(int[][] individu) {
2	Random random = new Random();
3	int popSize = individu.length;
4	int chromLength = individu[0].length;
5	int[][] offspringMutasi = new int[1][chromLength];
6	int batasMinParent, batasMaxParent;
7	int[][] parentTemp = individu.clone();
8	int[][] parent = new int[1][chromLength];
9	batasMinParent = 1;
10	batasMaxParent = popSize - 1;
11	int indexParent = random.nextInt((batasMaxParent -
12	batasMinParent + 1) + batasMinParent);
13	int[][] point = new int[banyakKelas][2];
14	boolean kondisi = true;
15	int point1 = 0, point2 = 0;
16	while (kondisi) {
17	point1 = random.nextInt(batasMaxIps12 - 1 -
18	batasMinBahasa10 + 1) + batasMinBahasa10;
19	point2 = random.nextInt(batasMaxIps12 - 1 -
20	batasMinBahasa10 + 1) + batasMinBahasa10;
21	if (point1 != point2) {
22	kondisi = false;
23	}
24	}
25	int random1 = 0, random2 = 0;
26	for (int i = 0; i < chromLength; i++) {
27	parent[0][i] = individu[indexParent][i];
28	offspringMutasi[0][i] = individu[indexParent][i];
29	random1 = parent[0][point1];
30	random2 = parent[0][point2];
31	}
32	offspringMutasi[0][point1] = random2;
33	offspringMutasi[0][point2] = random1;
34	return offspringMutasi;
35	}

5.2.4 Implementasi Constraint

5.2.4.1 Constraint Mata Pelajaran

1. Constraint 1

Constraint atau batasan pertama untuk penyusunan jadwal mata pelajaran adalah terdapat bentrok mata pelajaran pada satu kromosom. Proses

pengecekan *constraint* ini dilakukan dengan mengecek apakah gen muncul sebanyak lebih dari 1 kali yang ditunjukkan pada Tabel 5.8

Tabel 5.8 Implementasi kode program constraint mata pelajaran 1

```

1 public double[] Constraint1(int[][] individu) {
2     int popSize = individu.length;
3     int chromLength = individu[0].length;
4     double[] penalty1 = new double[popSize];
5     int[] bentrok = new int[popSize];
6     for (int k = 0; k < popSize; k++) {
7         for (int i = 0; i < chromLength; i++) {
8             for (int j = i + 1; j < chromLength; j++) {
9                 if (j != i && individu[k][j] == individu[k][i])
10 {
11                 bentrok[k] = bentrok[k] + 1;
12                 break;
13             }
14         }
15     }
16     penalty1[k] = bentrok[k] * 0.222;
17 }
18 return penalty1;
19 }
```

2. Constraint 2

Constraint atau batasan kedua untuk penyusunan jadwal mata pelajaran adalah apakah terdapat mata pelajaran dalam satu hari yang memiliki bobot lebih dari 6.29. Nilai 6.29 merupakan nilai bobot rata-rata yang dimiliki oleh bobot mata pelajaran. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah sesi 1 dan sesi 2 dalam satu hari memiliki bobot lebih dari 6.29 yang ditunjukkan pada Tabel 5.8

Tabel 5.9 Implementasi kode program constraint mata pelajaran 2

```

1 public double[] Constraint2(int[][] individu) {
2     Data data = new Data();
3     int popSize = individu.length;
4     double[] penalty2 = new double[individu.length];
5     int[] penalty = new int[popSize];
6     int bobotTengah = 6.29;
7     for (int i = 0; i < popSize; i++) {
8         if ((data.mapel[individu[i][0] - 1].bobotMapel +
9 data.mapel[individu[i][1] - 1].bobotMapel) > bobotTengah) {
10             penalty[i] = penalty[i] + 1;
11         } else if ((data.mapel[individu[i][2] - 1].bobotMapel
12 + data.mapel[individu[i][3] - 1].bobotMapel) > bobotTengah) {
13             penalty[i] = penalty[i] + 1;
14         } else if ((data.mapel[individu[i][4] - 1].bobotMapel
15 + data.mapel[individu[i][5]].bobotMapel) > bobotTengah) {
16             penalty[i] = penalty[i] + 1;
17         } else if ((data.mapel[individu[i][6] - 1].bobotMapel
18 + data.mapel[individu[i][7] - 1].bobotMapel) > bobotTengah) {
19             penalty[i] = penalty[i] + 1;
20         } else if ((data.mapel[individu[i][8] - 1].bobotMapel
21 + data.mapel[individu[i][9] - 1].bobotMapel) > bobotTengah) {
22             penalty[i] = penalty[i] + 1;
23         } else if ((data.mapel[individu[i][10] - 1].bobotMapel
24 + data.mapel[individu[i][11] - 1].bobotMapel) > bobotTengah) {
25             penalty[i] = penalty[i] + 1;
26         }
27     }
28     return penalty2;
29 }
```



```

26         } else if ((data.mapel[individu[i]][12] - 1].bobotMapel
27 + data.mapel[individu[i]][13] - 1].bobotMapel) > bobotTengah) {
28             penalty[i] = penalty[i] + 1;
29         } else if ((data.mapel[individu[i]][15] - 1].bobotMapel
30 + data.mapel[individu[i]][16] - 1].bobotMapel) > bobotTengah) {
31             penalty[i] = penalty[i] + 1;
32         } else if ((data.mapel[individu[i]][17] - 1].bobotMapel
33 + data.mapel[individu[i]][18] - 1].bobotMapel) > bobotTengah) {
34             penalty[i] = penalty[i] + 1;
35         } else if ((data.mapel[individu[i]][19] - 1].bobotMapel
36 + data.mapel[individu[i]][20] - 1].bobotMapel) > bobotTengah) {
37             penalty[i] = penalty[i] + 1;
38         } else if ((data.mapel[individu[i]][21] - 1].bobotMapel
39 + data.mapel[individu[i]][22] - 1].bobotMapel) > bobotTengah) {
40             penalty[i] = penalty[i] + 1;
41         } else if ((data.mapel[individu[i]][23] - 1].bobotMapel
42 + data.mapel[individu[i]][24] - 1].bobotMapel) > bobotTengah) {
43             penalty[i] = penalty[i] + 1;
44         } else if ((data.mapel[individu[i]][25] - 1].bobotMapel
45 + data.mapel[individu[i]][26] - 1].bobotMapel) > bobotTengah) {
46             penalty[i] = penalty[i] + 1;
47         } else if ((data.mapel[individu[i]][27] - 1].bobotMapel
48 + data.mapel[individu[i]][28] - 1].bobotMapel) > bobotTengah) {
49             penalty[i] = penalty[i] + 1;
50         } else if ((data.mapel[individu[i]][29] - 1].bobotMapel
51 + data.mapel[individu[i]][30] - 1].bobotMapel) > bobotTengah) {
52             penalty[i] = penalty[i] + 1;
53         } else if ((data.mapel[individu[i]][31] - 1].bobotMapel
54 + data.mapel[individu[i]][32] - 1].bobotMapel) > bobotTengah) {
55             penalty[i] = penalty[i] + 1;
56         } else if ((data.mapel[individu[i]][33] - 1].bobotMapel
57 + data.mapel[individu[i]][34] - 1].bobotMapel) > bobotTengah) {
58             penalty[i] = penalty[i] + 1;
59         } else if ((data.mapel[individu[i]][35] - 1].bobotMapel
60 + data.mapel[individu[i]][36] - 1].bobotMapel) > bobotTengah) {
61             penalty[i] = penalty[i] + 1;
62         } else if ((data.mapel[individu[i]][37] - 1].bobotMapel
63 + data.mapel[individu[i]][38] - 1].bobotMapel) > bobotTengah) {
64             penalty[i] = penalty[i] + 1;
65         } else if ((data.mapel[individu[i]][39] - 1].bobotMapel
66 + data.mapel[individu[i]][40] - 1].bobotMapel) > bobotTengah) {
67             penalty[i] = penalty[i] + 1;
68         } else if ((data.mapel[individu[i]][41] - 1].bobotMapel
69 + data.mapel[individu[i]][42] - 1].bobotMapel) > bobotTengah) {
70             penalty[i] = penalty[i] + 1;
71         } else if ((data.mapel[individu[i]][43] - 1].bobotMapel
72 + data.mapel[individu[i]][44] - 1].bobotMapel) > bobotTengah) {
73             penalty[i] = penalty[i] + 1;
74         } else if ((data.mapel[individu[i]][45] - 1].bobotMapel
75 + data.mapel[individu[i]][46] - 1].bobotMapel) > bobotTengah) {
76             penalty[i] = penalty[i] + 1;
77         } else if ((data.mapel[individu[i]][47] - 1].bobotMapel
78 + data.mapel[individu[i]][48] - 1].bobotMapel) > bobotTengah) {
79             penalty[i] = penalty[i] + 1;
80         } else if ((data.mapel[individu[i]][49] - 1].bobotMapel
81 + data.mapel[individu[i]][50] - 1].bobotMapel) > bobotTengah) {
82             penalty[i] = penalty[i] + 1;
83         } else if ((data.mapel[individu[i]][51] - 1].bobotMapel
84 + data.mapel[individu[i]][52] - 1].bobotMapel) > bobotTengah) {
85             penalty[i] = penalty[i] + 1;
86         } else if ((data.mapel[individu[i]][53] - 1].bobotMapel
87 + data.mapel[individu[i]][54] - 1].bobotMapel) > bobotTengah) {
88             penalty[i] = penalty[i] + 1;
89         } else if ((data.mapel[individu[i]][55] - 1].bobotMapel
90 + data.mapel[individu[i]][56] - 1].bobotMapel) > bobotTengah) {

```

```

91         penalty[i] = penalty[i] + 1;
92     } else if ((data.mapel[individu[i][58] - 1].bobotMapel
93 + data.mapel[individu[i][59] - 1].bobotMapel) > bobotTengah) {
94         penalty[i] = penalty[i] + 1;
95     } else if ((data.mapel[individu[i][60] - 1].bobotMapel
96 + data.mapel[individu[i][61] - 1].bobotMapel) > bobotTengah) {
97         penalty[i] = penalty[i] + 1;
98     } else if ((data.mapel[individu[i][62] - 1].bobotMapel
99 + data.mapel[individu[i][63] - 1].bobotMapel) > bobotTengah) {
100        penalty[i] = penalty[i] + 1;
101    } else if ((data.mapel[individu[i][64] - 1].bobotMapel
102 + data.mapel[individu[i][65] - 1].bobotMapel) > bobotTengah) {
103        penalty[i] = penalty[i] + 1;
104    } else if ((data.mapel[individu[i][66] - 1].bobotMapel
105 + data.mapel[individu[i][67] - 1].bobotMapel) > bobotTengah) {
106        penalty[i] = penalty[i] + 1;
107    } else if ((data.mapel[individu[i][68] - 1].bobotMapel
108 + data.mapel[individu[i][69] - 1].bobotMapel) > bobotTengah) {
109        penalty[i] = penalty[i] + 1;
110    } else if ((data.mapel[individu[i][70] - 1].bobotMapel
111 + data.mapel[individu[i][71] - 1].bobotMapel) > bobotTengah) {
112        penalty[i] = penalty[i] + 1;
113    } else if ((data.mapel[individu[i][72] - 1].bobotMapel
114 + data.mapel[individu[i][73] - 1].bobotMapel) > bobotTengah) {
115        penalty[i] = penalty[i] + 1;
116    } else if ((data.mapel[individu[i][74] - 1].bobotMapel
117 + data.mapel[individu[i][75] - 1].bobotMapel) > bobotTengah) {
118        penalty[i] = penalty[i] + 1;
119    } else if ((data.mapel[individu[i][76] - 1].bobotMapel
120 + data.mapel[individu[i][77] - 1].bobotMapel) > bobotTengah) {
121        penalty[i] = penalty[i] + 1;
122    } else if ((data.mapel[individu[i][78] - 1].bobotMapel
123 + data.mapel[individu[i][79] - 1].bobotMapel) > bobotTengah) {
124        penalty[i] = penalty[i] + 1;
125    } else if ((data.mapel[individu[i][80] - 1].bobotMapel
126 + data.mapel[individu[i][81] - 1].bobotMapel) > bobotTengah) {
127        penalty[i] = penalty[i] + 1;
128    } else if ((data.mapel[individu[i][82] - 1].bobotMapel
129 + data.mapel[individu[i][83] - 1].bobotMapel) > bobotTengah) {
130        penalty[i] = penalty[i] + 1;
131    } else if ((data.mapel[individu[i][84] - 1].bobotMapel
132 + data.mapel[individu[i][85] - 1].bobotMapel) > bobotTengah) {
133        penalty[i] = penalty[i] + 1;
134    } else if ((data.mapel[individu[i][86] - 1].bobotMapel
135 + data.mapel[individu[i][87] - 1].bobotMapel) > bobotTengah) {
136        penalty[i] = penalty[i] + 1;
137    } else if ((data.mapel[individu[i][88] - 1].bobotMapel
138 + data.mapel[individu[i][89] - 1].bobotMapel) > bobotTengah) {
139        penalty[i] = penalty[i] + 1;
140    } else if ((data.mapel[individu[i][90] - 1].bobotMapel
141 + data.mapel[individu[i][91] - 1].bobotMapel) > bobotTengah) {
142        penalty[i] = penalty[i] + 1;
143    } else if ((data.mapel[individu[i][92] - 1].bobotMapel
144 + data.mapel[individu[i][93] - 1].bobotMapel) > bobotTengah) {
145        penalty[i] = penalty[i] + 1;
146    } else if ((data.mapel[individu[i][94] - 1].bobotMapel
147 + data.mapel[individu[i][95] - 1].bobotMapel) > bobotTengah) {
148        penalty[i] = penalty[i] + 1;
149    } else if ((data.mapel[individu[i][96] - 1].bobotMapel
150 + data.mapel[individu[i][97] - 1].bobotMapel) > bobotTengah) {
151        penalty[i] = penalty[i] + 1;
152    } else if ((data.mapel[individu[i][98] - 1].bobotMapel
153 + data.mapel[individu[i][99] - 1].bobotMapel) > bobotTengah) {
154        penalty[i] = penalty[i] + 1;
155

```



```

156         } else if ((data.mapel[individu[i][101] - 1].bobotMapel
157 + data.mapel[individu[i][102] - 1].bobotMapel) > bobotTengah) {
158             penalty[i] = penalty[i] + 1;
159         } else if ((data.mapel[individu[i][103] - 1].bobotMapel
160 + data.mapel[individu[i][104] - 1].bobotMapel) > bobotTengah) {
161             penalty[i] = penalty[i] + 1;
162         } else if ((data.mapel[individu[i][105] - 1].bobotMapel
163 + data.mapel[individu[i][106] - 1].bobotMapel) > bobotTengah) {
164             penalty[i] = penalty[i] + 1;
165         } else if ((data.mapel[individu[i][107] - 1].bobotMapel
166 + data.mapel[individu[i][108] - 1].bobotMapel) > bobotTengah) {
167             penalty[i] = penalty[i] + 1;
168         } else if ((data.mapel[individu[i][109] - 1].bobotMapel
169 + data.mapel[individu[i][110] - 1].bobotMapel) > bobotTengah) {
170             penalty[i] = penalty[i] + 1;
171         } else if ((data.mapel[individu[i][111] - 1].bobotMapel
172 + data.mapel[individu[i][112] - 1].bobotMapel) > bobotTengah) {
173             penalty[i] = penalty[i] + 1;
174         } else if ((data.mapel[individu[i][113] - 1].bobotMapel
175 + data.mapel[individu[i][114] - 1].bobotMapel) > bobotTengah) {
176             penalty[i] = penalty[i] + 1;
177         } else if ((data.mapel[individu[i][115] - 1].bobotMapel
178 + data.mapel[individu[i][116] - 1].bobotMapel) > bobotTengah) {
179             penalty[i] = penalty[i] + 1;
180         } else if ((data.mapel[individu[i][117] - 1].bobotMapel
181 + data.mapel[individu[i][118] - 1].bobotMapel) > bobotTengah) {
182             penalty[i] = penalty[i] + 1;
183         } else if ((data.mapel[individu[i][119] - 1].bobotMapel
184 + data.mapel[individu[i][120] - 1].bobotMapel) > bobotTengah) {
185             penalty[i] = penalty[i] + 1;
186         } else if ((data.mapel[individu[i][121] - 1].bobotMapel
187 + data.mapel[individu[i][122] - 1].bobotMapel) > bobotTengah) {
188             penalty[i] = penalty[i] + 1;
189         } else if ((data.mapel[individu[i][123] - 1].bobotMapel
190 + data.mapel[individu[i][124] - 1].bobotMapel) > bobotTengah) {
191             penalty[i] = penalty[i] + 1;
192         } else if ((data.mapel[individu[i][125] - 1].bobotMapel
193 + data.mapel[individu[i][126] - 1].bobotMapel) > bobotTengah) {
194             penalty[i] = penalty[i] + 1;
195         } else if ((data.mapel[individu[i][127] - 1].bobotMapel
196 + data.mapel[individu[i][128] - 1].bobotMapel) > bobotTengah) {
197             penalty[i] = penalty[i] + 1;
198         }
199         penalty2[i] = penalty[i] * 0.222;
200     }
201     return penalty2;
202 }

```

3. Constraint 3

Constraint atau batasan ketiga untuk penyusunan jadwal mata pelajaran adalah apakah terdapat mata pelajaran pada sesi 2 yang memiliki bobot lebih besar dari sesi 1 dalam satu hari. Nilai 50 merupakan nilai bobot tengah yang dimiliki oleh mata pelajaran. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah sesi 2 memiliki bobot lebih besar dari sesi 1 yang ditunjukkan pada Tabel 5.10.

Tabel 5.10 Implementasi kode program constraint mata pelajaran 3

1	public double[] Constraint3(int[][] individu) {
2	Data data = new Data();

```

3      int popSize = individu.length;
4      int chromLength = individu[0].length;
5      double[] penalty3 = new double[popSize];
6      int[] penalty = new int[popSize];
7      for (int i = 0; i < popSize; i++) {
8          if (data.mapel[individu[i][0] - 1].bobotMapel <
9      data.mapel[individu[i][1] - 1].bobotMapel) {
10             penalty[i] = penalty[i] + 1;
11         } else if (data.mapel[individu[i][2] - 1].bobotMapel <
12      data.mapel[individu[i][3] - 1].bobotMapel) {
13             penalty[i] = penalty[i] + 1;
14         } else if (data.mapel[individu[i][4] - 1].bobotMapel <
15      data.mapel[individu[i][5] - 1].bobotMapel) {
16             penalty[i] = penalty[i] + 1;
17         } else if (data.mapel[individu[i][6] - 1].bobotMapel <
18      data.mapel[individu[i][7] - 1].bobotMapel) {
19             penalty[i] = penalty[i] + 1;
20         } else if (data.mapel[individu[i][8] - 1].bobotMapel <
21      data.mapel[individu[i][9] - 1].bobotMapel) {
22             penalty[i] = penalty[i] + 1;
23         } else if (data.mapel[individu[i][10] - 1].bobotMapel
24      < data.mapel[individu[i][11] - 1].bobotMapel) {
25             penalty[i] = penalty[i] + 1;
26         } else if (data.mapel[individu[i][12] - 1].bobotMapel
27      < data.mapel[individu[i][13] - 1].bobotMapel) {
28             penalty[i] = penalty[i] + 1;
29         } else if (data.mapel[individu[i][15] - 1].bobotMapel
30      < data.mapel[individu[i][16] - 1].bobotMapel) {
31             penalty[i] = penalty[i] + 1;
32         } else if (data.mapel[individu[i][17] - 1].bobotMapel
33      < data.mapel[individu[i][18] - 1].bobotMapel) {
34             penalty[i] = penalty[i] + 1;
35         } else if (data.mapel[individu[i][19] - 1].bobotMapel
36      < data.mapel[individu[i][20] - 1].bobotMapel) {
37             penalty[i] = penalty[i] + 1;
38         } else if (data.mapel[individu[i][21] - 1].bobotMapel
39      < data.mapel[individu[i][22] - 1].bobotMapel) {
40             penalty[i] = penalty[i] + 1;
41         } else if (data.mapel[individu[i][23] - 1].bobotMapel
42      < data.mapel[individu[i][24] - 1].bobotMapel) {
43             penalty[i] = penalty[i] + 1;
44         } else if (data.mapel[individu[i][25] - 1].bobotMapel
45      < data.mapel[individu[i][26] - 1].bobotMapel) {
46             penalty[i] = penalty[i] + 1;
47         } else if (data.mapel[individu[i][27] - 1].bobotMapel
48      < data.mapel[individu[i][28] - 1].bobotMapel) {
49             penalty[i] = penalty[i] + 1;
50         } else if (data.mapel[individu[i][29] - 1].bobotMapel
51      < data.mapel[individu[i][30] - 1].bobotMapel) {
52             penalty[i] = penalty[i] + 1;
53         } else if (data.mapel[individu[i][31] - 1].bobotMapel
54      < data.mapel[individu[i][32] - 1].bobotMapel) {
55             penalty[i] = penalty[i] + 1;
56         } else if (data.mapel[individu[i][33] - 1].bobotMapel
57      < data.mapel[individu[i][34] - 1].bobotMapel) {
58             penalty[i] = penalty[i] + 1;
59         } else if (data.mapel[individu[i][35] - 1].bobotMapel
60      < data.mapel[individu[i][36] - 1].bobotMapel) {
61             penalty[i] = penalty[i] + 1;
62         } else if (data.mapel[individu[i][37] - 1].bobotMapel
63      < data.mapel[individu[i][38] - 1].bobotMapel) {
64             penalty[i] = penalty[i] + 1;
65         } else if (data.mapel[individu[i][39] - 1].bobotMapel
66      < data.mapel[individu[i][40] - 1].bobotMapel) {
67             penalty[i] = penalty[i] + 1;

```

```

68         } else if (data.mapel[individu[i][41] - 1].bobotMapel
69 < data.mapel[individu[i][42] - 1].bobotMapel) {
70             penalty[i] = penalty[i] + 1;
71         } else if (data.mapel[individu[i][43] - 1].bobotMapel
72 < data.mapel[individu[i][44] - 1].bobotMapel) {
73             penalty[i] = penalty[i] + 1;
74         } else if (data.mapel[individu[i][45] - 1].bobotMapel
75 < data.mapel[individu[i][46] - 1].bobotMapel) {
76             penalty[i] = penalty[i] + 1;
77         } else if (data.mapel[individu[i][47] - 1].bobotMapel
78 < data.mapel[individu[i][48] - 1].bobotMapel) {
79             penalty[i] = penalty[i] + 1;
80         } else if (data.mapel[individu[i][49] - 1].bobotMapel
81 < data.mapel[individu[i][50] - 1].bobotMapel) {
82             penalty[i] = penalty[i] + 1;
83         } else if (data.mapel[individu[i][51] - 1].bobotMapel
84 < data.mapel[individu[i][52] - 1].bobotMapel) {
85             penalty[i] = penalty[i] + 1;
86         } else if (data.mapel[individu[i][53] - 1].bobotMapel
87 < data.mapel[individu[i][54] - 1].bobotMapel) {
88             penalty[i] = penalty[i] + 1;
89         } else if (data.mapel[individu[i][55] - 1].bobotMapel
90 < data.mapel[individu[i][56] - 1].bobotMapel) {
91             penalty[i] = penalty[i] + 1;
92         } else if (data.mapel[individu[i][58] - 1].bobotMapel
93 < data.mapel[individu[i][59] - 1].bobotMapel) {
94             penalty[i] = penalty[i] + 1;
95         } else if (data.mapel[individu[i][60] - 1].bobotMapel
96 < data.mapel[individu[i][61] - 1].bobotMapel) {
97             penalty[i] = penalty[i] + 1;
98         } else if (data.mapel[individu[i][62] - 1].bobotMapel
99 < data.mapel[individu[i][63] - 1].bobotMapel) {
100             penalty[i] = penalty[i] + 1;
101         } else if (data.mapel[individu[i][64] - 1].bobotMapel
102 < data.mapel[individu[i][65] - 1].bobotMapel) {
103             penalty[i] = penalty[i] + 1;
104         } else if (data.mapel[individu[i][66] - 1].bobotMapel
105 < data.mapel[individu[i][67] - 1].bobotMapel) {
106             penalty[i] = penalty[i] + 1;
107         } else if (data.mapel[individu[i][68] - 1].bobotMapel
108 < data.mapel[individu[i][69] - 1].bobotMapel) {
109             penalty[i] = penalty[i] + 1;
110         } else if (data.mapel[individu[i][70] - 1].bobotMapel
111 < data.mapel[individu[i][71] - 1].bobotMapel) {
112             penalty[i] = penalty[i] + 1;
113         } else if (data.mapel[individu[i][72] - 1].bobotMapel
114 < data.mapel[individu[i][73] - 1].bobotMapel) {
115             penalty[i] = penalty[i] + 1;
116         } else if (data.mapel[individu[i][74] - 1].bobotMapel
117 < data.mapel[individu[i][75] - 1].bobotMapel) {
118             penalty[i] = penalty[i] + 1;
119         } else if (data.mapel[individu[i][76] - 1].bobotMapel
120 < data.mapel[individu[i][77] - 1].bobotMapel) {
121             penalty[i] = penalty[i] + 1;
122         } else if (data.mapel[individu[i][78] - 1].bobotMapel
123 < data.mapel[individu[i][79] - 1].bobotMapel) {
124             penalty[i] = penalty[i] + 1;
125         } else if (data.mapel[individu[i][80] - 1].bobotMapel
126 < data.mapel[individu[i][81] - 1].bobotMapel) {
127             penalty[i] = penalty[i] + 1;
128         } else if (data.mapel[individu[i][82] - 1].bobotMapel
129 < data.mapel[individu[i][83] - 1].bobotMapel) {
130             penalty[i] = penalty[i] + 1;
131         } else if (data.mapel[individu[i][84] - 1].bobotMapel
132 < data.mapel[individu[i][85] - 1].bobotMapel) {

```

```

133         penalty[i] = penalty[i] + 1;
134     } else if (data.mapel[individu[i][86] - 1].bobotMapel
135 < data.mapel[individu[i][87] - 1].bobotMapel) {
136         penalty[i] = penalty[i] + 1;
137     } else if (data.mapel[individu[i][88] - 1].bobotMapel
138 < data.mapel[individu[i][89] - 1].bobotMapel) {
139         penalty[i] = penalty[i] + 1;
140     } else if (data.mapel[individu[i][90] - 1].bobotMapel
141 < data.mapel[individu[i][91] - 1].bobotMapel) {
142         penalty[i] = penalty[i] + 1;
143     } else if (data.mapel[individu[i][92] - 1].bobotMapel
144 < data.mapel[individu[i][93] - 1].bobotMapel) {
145         penalty[i] = penalty[i] + 1;
146     } else if (data.mapel[individu[i][94] - 1].bobotMapel
147 < data.mapel[individu[i][95] - 1].bobotMapel) {
148         penalty[i] = penalty[i] + 1;
149     } else if (data.mapel[individu[i][96] - 1].bobotMapel
150 < data.mapel[individu[i][97] - 1].bobotMapel) {
151         penalty[i] = penalty[i] + 1;
152     } else if (data.mapel[individu[i][98] - 1].bobotMapel
153 < data.mapel[individu[i][99] - 1].bobotMapel) {
154         penalty[i] = penalty[i] + 1;
155     } else if (data.mapel[individu[i][101] - 1].bobotMapel
156 < data.mapel[individu[i][102] - 1].bobotMapel) {
157         penalty[i] = penalty[i] + 1;
158     } else if (data.mapel[individu[i][103] - 1].bobotMapel
159 < data.mapel[individu[i][104] - 1].bobotMapel) {
160         penalty[i] = penalty[i] + 1;
161     } else if (data.mapel[individu[i][105] - 1].bobotMapel
162 < data.mapel[individu[i][106] - 1].bobotMapel) {
163         penalty[i] = penalty[i] + 1;
164     } else if (data.mapel[individu[i][107] - 1].bobotMapel
165 < data.mapel[individu[i][108] - 1].bobotMapel) {
166         penalty[i] = penalty[i] + 1;
167     } else if (data.mapel[individu[i][109] - 1].bobotMapel
168 < data.mapel[individu[i][110] - 1].bobotMapel) {
169         penalty[i] = penalty[i] + 1;
170     } else if (data.mapel[individu[i][111] - 1].bobotMapel
171 < data.mapel[individu[i][112] - 1].bobotMapel) {
172         penalty[i] = penalty[i] + 1;
173     } else if (data.mapel[individu[i][113] - 1].bobotMapel
174 < data.mapel[individu[i][114] - 1].bobotMapel) {
175         penalty[i] = penalty[i] + 1;
176     } else if (data.mapel[individu[i][115] - 1].bobotMapel
177 < data.mapel[individu[i][116] - 1].bobotMapel) {
178         penalty[i] = penalty[i] + 1;
179     } else if (data.mapel[individu[i][117] - 1].bobotMapel
180 < data.mapel[individu[i][118] - 1].bobotMapel) {
181         penalty[i] = penalty[i] + 1;
182     } else if (data.mapel[individu[i][119] - 1].bobotMapel
183 < data.mapel[individu[i][120] - 1].bobotMapel) {
184         penalty[i] = penalty[i] + 1;
185     } else if (data.mapel[individu[i][121] - 1].bobotMapel
186 < data.mapel[individu[i][122] - 1].bobotMapel) {
187         penalty[i] = penalty[i] + 1;
188     } else if (data.mapel[individu[i][123] - 1].bobotMapel
189 < data.mapel[individu[i][124] - 1].bobotMapel) {
190         penalty[i] = penalty[i] + 1;
191     } else if (data.mapel[individu[i][125] - 1].bobotMapel
192 < data.mapel[individu[i][126] - 1].bobotMapel) {
193         penalty[i] = penalty[i] + 1;
194     } else if (data.mapel[individu[i][127] - 1].bobotMapel
195 < data.mapel[individu[i][128] - 1].bobotMapel) {
196         penalty[i] = penalty[i] + 1;
197     }

```

198	penalty3[i] = penalty[i] * 0.222;
199	}
200	return penalty3;
201	}

4. Constraint 4

Constraint atau batasan keempat untuk penyusunan jadwal mata pelajaran adalah apakah terdapat mata pelajaran ujian nasional pada sesi 2. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah pada sesi 2 memiliki mata pelajaran yang termasuk mata pelajaran ujian nasional yang ditunjukkan pada Tabel 5.11

Tabel 5.11 Implementasi kode program constraint mata pelajaran 4

1	public double[] Constraint4(int[][] individu) {
2	Data data = new Data();
3	int popSize = individu.length;
4	int chromLength = individu[0].length;
5	double[] penalty4 = new double[popSize];
6	int[] penalty = new int[popSize];
7	for (int i = 0; i < popSize; i++) {
8	for (int j = 0; j < chromLength; j++) {
9	if (j <= 15) {
10	if (j % 2 == 1 && data.mapel[individu[i][j] -
11	1].keterangan.equalsIgnoreCase("UN")) {
12	penalty[i] = penalty[i] + 1;
13	}
14	} else if ((j > 15) && (j <= 29)) {
15	if (j % 2 == 0 && data.mapel[individu[i][j] -
16	1].keterangan.equalsIgnoreCase("UN")) {
17	penalty[i] = penalty[i] + 1;
18	}
19	} else if ((j > 29) && (j <= 43)) {
20	if (j % 2 == 0 && data.mapel[individu[i][j] -
21	1].keterangan.equalsIgnoreCase("UN")) {
22	penalty[i] = penalty[i] + 1;
23	}
24	} else if ((j > 43) && (j <= 58)) {
25	if (j % 2 == 0 && data.mapel[individu[i][j] -
26	1].keterangan.equalsIgnoreCase("UN")) {
27	penalty[i] = penalty[i] + 1;
28	}
29	} else if ((j > 58) && (j <= 72)) {
30	if (j % 2 == 1 && data.mapel[individu[i][j] -
31	1].keterangan.equalsIgnoreCase("UN")) {
32	penalty[i] = penalty[i] + 1;
33	}
34	} else if ((j > 72) && (j <= 86)) {
35	if (j % 2 == 0 && data.mapel[individu[i][j] -
36	1].keterangan.equalsIgnoreCase("UN")) {
37	penalty[i] = penalty[i] + 1;
38	}
39	} else if ((j > 86) && (j <= 101)) {
40	if (j % 2 == 1 && data.mapel[individu[i][j] -
41	1].keterangan.equalsIgnoreCase("UN")) {
42	penalty[i] = penalty[i] + 1;
43	}
44	} else if ((j > 101) && (j <= 115)) {
45	if (j % 2 == 0 && data.mapel[individu[i][j] -
46	1].keterangan.equalsIgnoreCase("UN")) {
47	penalty[i] = penalty[i] + 1;


```

48         }
49         } else if (j > 115) {
50             if (j % 2 == 0 && data.mapel[individu[i][j] -
51 1].keterangan.equalsIgnoreCase("UN")) {
52                 penalty[i] = penalty[i] + 1;
53             }
54         }
55     }
56     penalty4[i] = penalty[i] * 0.111;
57 }
58 return penalty4;
59 }

```

5.2.4.2 Constraint Pengawas

1. Constraint 1

Constraint atau batasan pertama untuk penyusunan jadwal pengawas adalah apakah terdapat bentrok jadwal mengawas pada tiap sesi. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah pada tiap sesi terdapat mata pelajaran yang muncul lebih dari satu kali yang ditunjukkan pada Tabel 5.13.

Tabel 5.12 Implementasi kode program constraint pengawas 1

```

1 public double[] ConstraintPengawas1(int[][] individu) {
2     Data data = new Data();
3     double[] penalty1 = new double[individu.length];
4     double[] bentrok = new double[individu.length];
5     for (int i = 0; i < individu.length; i++) {
6         for (int j = 0; j < individu[0].length; j++) {
7             for (int k = i + 1; k < individu[0].length; k++) {
8                 //SESI 1
9                 if (j <= 29) { //hari1
10                     if (k != j &&
11 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
12 [individu[i][j]].kodePengawas)) {
13                         bentrok[i] = bentrok[i] + 1;
14                     }
15                 } else if (j > 29 && j <= 59) { //hari2
16                     if (k != j &&
17 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
18 [individu[i][j]].kodePengawas)) {
19                         bentrok[i] = bentrok[i] + 1;
20                     }
21                 } else if (j > 59 && j <= 89) { //hari3
22                     if (k != j &&
23 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
24 [individu[i][j]].kodePengawas)) {
25                         bentrok[i] = bentrok[i] + 1;
26                     }
27                 } else if (j > 89 && j <= 119) { //hari4
28                     if (k != j &&
29 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
30 [individu[i][j]].kodePengawas)) {
31                         bentrok[i] = bentrok[i] + 1;
32                     }
33                 } else if (j > 119 && j <= 149) { //hari5
34                     if (k != j &&
35 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
36 [individu[i][j]].kodePengawas)) {

```

```

37         bentrok[i] = bentrok[i] + 1;
38     }
39     } else if (j > 149 && j <= 179) { //hari6
40         if (k != j &&
41 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
42 [individu[i][j]].kodePengawas)) {
43         bentrok[i] = bentrok[i] + 1;
44     }
45     } else if (j > 179 && j <= 209) { //hari7
46         if (k != j &&
47 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
48 [individu[i][j]].kodePengawas)) {
49         bentrok[i] = bentrok[i] + 1;
50     }
51     } else if (j > 209 && j <= 219) { //hari8
52         if (k != j &&
53 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
54 [individu[i][j]].kodePengawas)) {
55         bentrok[i] = bentrok[i] + 1;
56     }
57     //SESI 2
58     } else if (j > 219 && j <= 249) {
59         if (k != j &&
60 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
61 [individu[i][j]].kodePengawas)) {
62         bentrok[i] = bentrok[i] + 1;
63     }
64     } else if (j > 249 && j <= 279) {
65         if (k != j &&
66 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
67 [individu[i][j]].kodePengawas)) {
68         bentrok[i] = bentrok[i] + 1;
69     }
70     } else if (j > 279 && j <= 309) {
71         if (k != j &&
72 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
73 [individu[i][j]].kodePengawas)) {
74         bentrok[i] = bentrok[i] + 1;
75     }
76     } else if (j > 309 && j <= 339) {
77         if (k != j &&
78 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
79 [individu[i][j]].kodePengawas)) {
80         bentrok[i] = bentrok[i] + 1;
81     }
82     } else if (j > 339 && j <= 369) {
83         if (k != j &&
84 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
85 [individu[i][j]].kodePengawas)) {
86         bentrok[i] = bentrok[i] + 1;
87     }
88     } else if (j > 369 && j <= 399) {
89         if (k != j &&
90 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
91 [individu[i][j]].kodePengawas)) {
92         bentrok[i] = bentrok[i] + 1;
93     }
94     } else if (j > 399) {
95         if (k != j &&
96 data.guru[individu[i][k]].kodePengawas.equalsIgnoreCase(data.guru
97 [individu[i][j]].kodePengawas)) {
98         bentrok[i] = bentrok[i] + 1;
99     }
100 }
101 }

```


102	}
103	penalty1[i] = bentrok[i] * 0.5;
104	}
105	return penalty1;
106	}

2. Constraint 2

Constraint atau batasan kedua untuk penyusunan jadwal pengawas adalah apakah terdapat pengawas yang menjaga untuk mata pelajaran yang tidak dikuasai. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah pada terdapat pengawas yang tidak mengawas sesuai mata pelajaran pada penjadwalan mata pelajaran yang ditunjukkan pada Tabel 5.14.

Tabel 5.13 Implementasi kode program constraint pengawas 2

1	public double[] ConstraintPengawas2(int[][] individu, int[][]
2	indvPengawas) {
3	Data data = new Data();
4	double[] penalty2 = new double[indvPengawas.length];
5	double[] penalty = new double[indvPengawas.length];
6	for (int i = 0; i < indvPengawas.length; i++) {
7	for (int j = 0; j < indvPengawas[0].length; j++) {
8	for (int k = 0; k < individu[0].length; k++) {
9	if (j == 0 && k == 0) { // x bahasa hari 1
10	if (!data.guru[indvPengawas[i][j]
11	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
12	1].kodeMapel)) {
13	penalty[i] = penalty[i] + 1;
14	}
15	} else if (j == 1 && k == 15) {
16	if (!data.guru[indvPengawas[i][j]
17	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
18	1].kodeMapel)) {
19	penalty[i] = penalty[i] + 1;
20	}
21	} else if (j == 2 && k == 29) {
22	if (!data.guru[indvPengawas[i][j]
23	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
24	1].kodeMapel)) {
25	penalty[i] = penalty[i] + 1;
26	}
27	} else if ((j > 2 && j <= 8) && k == 45) {
28	if (!data.guru[indvPengawas[i][j]
29	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
30	1].kodeMapel)) {
31	penalty[i] = penalty[i] + 1;
32	}
33	} else if ((j > 8 && j <= 14) && k == 59) {
34	if (!data.guru[indvPengawas[i][j]
35	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
36	1].kodeMapel)) {
37	penalty[i] = penalty[i] + 1;
38	}
39	} else if ((j > 14 && j <= 20) && k == 72) {
40	if (!data.guru[indvPengawas[i][j]
41	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
42	1].kodeMapel)) {
43	penalty[i] = penalty[i] + 1;
44	}
45	} else if ((j > 20 && j <= 23) && k == 86) {
46	

```

47         if (!data.guru[indvPengawas[i][j]) -
48 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
49 1].kodeMapel)) {
50             penalty[i] = penalty[i] + 1;
51         }
52     } else if ((j > 23 && j <= 26) && k == 101) {
53         if (!data.guru[indvPengawas[i][j]) -
54 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
55 1].kodeMapel)) {
56             penalty[i] = penalty[i] + 1;
57         }
58     } else if ((j > 26 && j <= 29) && k == 115) {
59         if (!data.guru[indvPengawas[i][j]) -
60 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
61 1].kodeMapel)) {
62             penalty[i] = penalty[i] + 1;
63         }
64     } else if (j == 30 && k == 2) {
65         if (!data.guru[indvPengawas[i][j]) -
66 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
67 1].kodeMapel)) {
68             penalty[i] = penalty[i] + 1;
69         }
70     } else if (j == 31 && k == 17) {
71         if (!data.guru[indvPengawas[i][j]) -
72 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
73 1].kodeMapel)) {
74             penalty[i] = penalty[i] + 1;
75         }
76     } else if (j == 32 && k == 31) {
77         if (!data.guru[indvPengawas[i][j]) -
78 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
79 1].kodeMapel)) {
80             penalty[i] = penalty[i] + 1;
81         }
82     } else if ((j > 32 && j <= 38) && k == 47) {
83         if (!data.guru[indvPengawas[i][j]) -
84 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
85 1].kodeMapel)) {
86             penalty[i] = penalty[i] + 1;
87         }
88     } else if ((j > 39 && j <= 44) && k == 61) {
89         if (!data.guru[indvPengawas[i][j]) -
90 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
91 1].kodeMapel)) {
92             penalty[i] = penalty[i] + 1;
93         }
94     } else if ((j > 44 && j <= 50) && k == 74) {
95         if (!data.guru[indvPengawas[i][j]) -
96 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
97 1].kodeMapel)) {
98             penalty[i] = penalty[i] + 1;
99         }
100     } else if ((j > 50 && j <= 53) && k == 88) {
101         if (!data.guru[indvPengawas[i][j]) -
102 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
103 1].kodeMapel)) {
104             penalty[i] = penalty[i] + 1;
105         }
106     } else if ((j > 53 && j <= 56) && k == 103) {
107         if (!data.guru[indvPengawas[i][j]) -
108 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
109 1].kodeMapel)) {
110             penalty[i] = penalty[i] + 1;
111         }

```

```

112         } else if ((j > 56 && j <= 59) && k == 117) {
113             if (!data.guru[indvPengawas[i][j] -
114 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
115 1].kodeMapel)) {
116                 penalty[i] = penalty[i] + 1;
117             }
118         } else if (j == 60 && k == 4) {
119             if (!data.guru[indvPengawas[i][j] -
120 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
121 1].kodeMapel)) {
122                 penalty[i] = penalty[i] + 1;
123             }
124         } else if (j == 61 && k == 19) {
125             if (!data.guru[indvPengawas[i][j] -
126 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
127 1].kodeMapel)) {
128                 penalty[i] = penalty[i] + 1;
129             }
130         } else if (j == 62 && k == 35) {
131             if (!data.guru[indvPengawas[i][j] -
132 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
133 1].kodeMapel)) {
134                 penalty[i] = penalty[i] + 1;
135             }
136         } else if ((j > 62 && j <= 68) && k == 49) {
137             if (!data.guru[indvPengawas[i][j] -
138 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
139 1].kodeMapel)) {
140                 penalty[i] = penalty[i] + 1;
141             }
142         } else if ((j > 68 && j <= 74) && k == 63) {
143             if (!data.guru[indvPengawas[i][j] -
144 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
145 1].kodeMapel)) {
146                 penalty[i] = penalty[i] + 1;
147             }
148         } else if ((j > 74 && j <= 80) && k == 76) {
149             if (!data.guru[indvPengawas[i][j] -
150 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
151 1].kodeMapel)) {
152                 penalty[i] = penalty[i] + 1;
153             }
154         } else if ((j > 80 && j <= 83) && k == 90) {
155             if (!data.guru[indvPengawas[i][j] -
156 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
157 1].kodeMapel)) {
158                 penalty[i] = penalty[i] + 1;
159             }
160         } else if ((j > 83 && j <= 86) && k == 105) {
161             if (!data.guru[indvPengawas[i][j] -
162 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
163 1].kodeMapel)) {
164                 penalty[i] = penalty[i] + 1;
165             }
166         } else if ((j > 86 && j <= 89) && k == 119) {
167             if (!data.guru[indvPengawas[i][j] -
168 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
169 1].kodeMapel)) {
170                 penalty[i] = penalty[i] + 1;
171             }
172         } else if (j == 90 && k == 6) {
173             if (!data.guru[indvPengawas[i][j] -
174 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
175 1].kodeMapel)) {
176                 penalty[i] = penalty[i] + 1;

```

```

177         }
178         } else if (j == 91 && k == 21) {
179             if (!data.guru[indvPengawas[i][j] -
180 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
181 1].kodeMapel)) {
182                 penalty[i] = penalty[i] + 1;
183             }
184         } else if (j == 92 && k == 37) {
185             if (!data.guru[indvPengawas[i][j] -
186 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
187 1].kodeMapel)) {
188                 penalty[i] = penalty[i] + 1;
189             }
190         } else if ((j > 92 && j <= 98) && k == 51) {
191             if (!data.guru[indvPengawas[i][j] -
192 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
193 1].kodeMapel)) {
194                 penalty[i] = penalty[i] + 1;
195             }
196         } else if ((j > 98 && j <= 104) && k == 65) {
197             if (!data.guru[indvPengawas[i][j] -
198 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
199 1].kodeMapel)) {
200                 penalty[i] = penalty[i] + 1;
201             }
202         } else if ((j > 104 && j <= 110) && k == 78) {
203             if (!data.guru[indvPengawas[i][j] -
204 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
205 1].kodeMapel)) {
206                 penalty[i] = penalty[i] + 1;
207             }
208         } else if ((j > 110 && j <= 113) && k == 92) {
209             if (!data.guru[indvPengawas[i][j] -
210 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
211 1].kodeMapel)) {
212                 penalty[i] = penalty[i] + 1;
213             }
214         } else if ((j > 113 && j <= 116) && k == 107) {
215             if (!data.guru[indvPengawas[i][j] -
216 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
217 1].kodeMapel)) {
218                 penalty[i] = penalty[i] + 1;
219             }
220         } else if ((j > 116 && j <= 119) && k == 121) {
221             if (!data.guru[indvPengawas[i][j] -
222 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
223 1].kodeMapel)) {
224                 penalty[i] = penalty[i] + 1;
225             }
226         } else if (j == 120 && k == 8) {
227             if (!data.guru[indvPengawas[i][j] -
228 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
229 1].kodeMapel)) {
230                 penalty[i] = penalty[i] + 1;
231             }
232         } else if (j == 121 && k == 23) {
233             if (!data.guru[indvPengawas[i][j] -
234 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
235 1].kodeMapel)) {
236                 penalty[i] = penalty[i] + 1;
237             }
238         } else if (j == 122 && k == 39) {
239             if (!data.guru[indvPengawas[i][j] -
240 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
241 1].kodeMapel)) {

```

```

243         penalty[i] = penalty[i] + 1;
244     }
245     } else if ((j > 122 && j <= 128) && k == 53) {
246         if (!data.guru[indvPengawas[i][j]) -
247 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
248 1].kodeMapel)) {
249             penalty[i] = penalty[i] + 1;
250         }
251     } else if ((j > 128 && j <= 134) && k == 67) {
252         if (!data.guru[indvPengawas[i][j]) -
253 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
254 1].kodeMapel)) {
255             penalty[i] = penalty[i] + 1;
256         }
257     } else if ((j > 134 && j <= 140) && k == 80) {
258         if (!data.guru[indvPengawas[i][j]) -
259 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
260 1].kodeMapel)) {
261             penalty[i] = penalty[i] + 1;
262         }
263     } else if ((j > 140 && j <= 143) && k == 94) {
264         if (!data.guru[indvPengawas[i][j]) -
265 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
266 1].kodeMapel)) {
267             penalty[i] = penalty[i] + 1;
268         }
269     } else if ((j > 143 && j <= 146) && k == 109) {
270         if (!data.guru[indvPengawas[i][j]) -
271 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
272 1].kodeMapel)) {
273             penalty[i] = penalty[i] + 1;
274         }
275     } else if ((j > 146 && j <= 149) && k == 123) {
276         if (!data.guru[indvPengawas[i][j]) -
277 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
278 1].kodeMapel)) {
279             penalty[i] = penalty[i] + 1;
280         }
281     } else if (j == 150 && k == 10) {
282         if (!data.guru[indvPengawas[i][j]) -
283 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
284 1].kodeMapel)) {
285             penalty[i] = penalty[i] + 1;
286         }
287     } else if (j == 151 && k == 25) {
288         if (!data.guru[indvPengawas[i][j]) -
289 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
290 1].kodeMapel)) {
291             penalty[i] = penalty[i] + 1;
292         }
293     } else if (j == 152 && k == 39) {
294         if (!data.guru[indvPengawas[i][j]) -
295 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
296 1].kodeMapel)) {
297             penalty[i] = penalty[i] + 1;
298         }
299     } else if ((j > 152 && j <= 158) && k == 55) {
300         if (!data.guru[indvPengawas[i][j]) -
301 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
302 1].kodeMapel)) {
303             penalty[i] = penalty[i] + 1;
304         }
305     } else if ((j > 158 && j <= 164) && k == 69) {
306
307

```

```

308         if (!data.guru[indvPengawas[i][j]) -
309 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
310 1].kodeMapel)) {
311             penalty[i] = penalty[i] + 1;
312         }
313     } else if ((j > 164 && j <= 170) && k == 82) {
314         if (!data.guru[indvPengawas[i][j]) -
315 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
316 1].kodeMapel)) {
317             penalty[i] = penalty[i] + 1;
318         }
319     } else if ((j > 170 && j <= 173) && k == 96) {
320         if (!data.guru[indvPengawas[i][j]) -
321 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
322 1].kodeMapel)) {
323             penalty[i] = penalty[i] + 1;
324         }
325     } else if ((j > 173 && j <= 176) && k == 111) {
326         if (!data.guru[indvPengawas[i][j]) -
327 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
328 1].kodeMapel)) {
329             penalty[i] = penalty[i] + 1;
330         }
331     } else if ((j > 176 && j <= 179) && k == 125) {
332         if (!data.guru[indvPengawas[i][j]) -
333 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
334 1].kodeMapel)) {
335             penalty[i] = penalty[i] + 1;
336         }
337     } else if (j == 180 && k == 12) {
338         if (!data.guru[indvPengawas[i][j]) -
339 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
340 1].kodeMapel)) {
341             penalty[i] = penalty[i] + 1;
342         }
343     } else if (j == 181 && k == 27) {
344         if (!data.guru[indvPengawas[i][j]) -
345 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
346 1].kodeMapel)) {
347             penalty[i] = penalty[i] + 1;
348         }
349     } else if (j == 182 && k == 43) {
350         if (!data.guru[indvPengawas[i][j]) -
351 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
352 1].kodeMapel)) {
353             penalty[i] = penalty[i] + 1;
354         }
355     } else if ((j > 182 && j <= 188) && k == 57) {
356         if (!data.guru[indvPengawas[i][j]) -
357 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
358 1].kodeMapel)) {
359             penalty[i] = penalty[i] + 1;
360         }
361     } else if ((j > 188 && j <= 194) && k == 71) {
362         if (!data.guru[indvPengawas[i][j]) -
363 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
364 1].kodeMapel)) {
365             penalty[i] = penalty[i] + 1;
366         }
367     } else if ((j > 194 && j <= 200) && k == 84) {
368         if (!data.guru[indvPengawas[i][j]) -
369 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
370 1].kodeMapel)) {
371             penalty[i] = penalty[i] + 1;
372         }

```



```

373         } else if ((j > 200 && j <= 203) && k == 98) {
374             if (!data.guru[indvPengawas[i][j] -
375 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
376 1].kodeMapel)) {
377                 penalty[i] = penalty[i] + 1;
378             }
379         } else if ((j > 203 && j <= 206) && k == 113) {
380             if (!data.guru[indvPengawas[i][j] -
381 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
382 1].kodeMapel)) {
383                 penalty[i] = penalty[i] + 1;
384             }
385         } else if ((j > 206 && j <= 209) && k == 127) {
386             if (!data.guru[indvPengawas[i][j] -
387 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
388 1].kodeMapel)) {
389                 penalty[i] = penalty[i] + 1;
390             }
391         } else if (j == 210 && k == 14) {
392             if (!data.guru[indvPengawas[i][j] -
393 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
394 1].kodeMapel)) {
395                 penalty[i] = penalty[i] + 1;
396             }
397         } else if ((j > 210 && j <= 216) && k == 59) {
398             if (!data.guru[indvPengawas[i][j] -
399 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
400 1].kodeMapel)) {
401                 penalty[i] = penalty[i] + 1;
402             }
403         } else if ((j > 216 && j <= 219) && k == 100) {
404             if (!data.guru[indvPengawas[i][j] -
405 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
406 1].kodeMapel)) {
407                 penalty[i] = penalty[i] + 1;
408             }
409         } else if (j == 220 && k == 1) {
410             if (!data.guru[indvPengawas[i][j] -
411 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
412 1].kodeMapel)) {
413                 penalty[i] = penalty[i] + 1;
414             }
415         } else if (j == 221 && k == 16) {
416             if (!data.guru[indvPengawas[i][j] -
417 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
418 1].kodeMapel)) {
419                 penalty[i] = penalty[i] + 1;
420             }
421         } else if (j == 222 && k == 30) {
422             if (!data.guru[indvPengawas[i][j] -
423 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
424 1].kodeMapel)) {
425                 penalty[i] = penalty[i] + 1;
426             }
427         } else if ((j > 222 && j <= 228) && k == 46) {
428             if (!data.guru[indvPengawas[i][j] -
429 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
430 1].kodeMapel)) {
431                 penalty[i] = penalty[i] + 1;
432             }
433         } else if ((j > 228 && j <= 234) && k == 60) {
434             if (!data.guru[indvPengawas[i][j] -
435 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
436 1].kodeMapel)) {
437                 penalty[i] = penalty[i] + 1;

```



```

438         }
439         } else if ((j > 234 && j <= 240) && k == 73) {
440             if (!data.guru[indvPengawas[i][j]) -
441 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
442 1].kodeMapel)) {
443                 penalty[i] = penalty[i] + 1;
444             }
445         } else if ((j > 240 && j <= 243) && k == 87) {
446             if (!data.guru[indvPengawas[i][j]) -
447 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
448 1].kodeMapel)) {
449                 penalty[i] = penalty[i] + 1;
450             }
451         } else if ((j > 243 && j <= 246) && k == 102) {
452             if (!data.guru[indvPengawas[i][j]) -
453 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
454 1].kodeMapel)) {
455                 penalty[i] = penalty[i] + 1;
456             }
457         } else if ((j > 246 && j <= 249) && k == 116) {
458             if (!data.guru[indvPengawas[i][j]) -
459 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
460 1].kodeMapel)) {
461                 penalty[i] = penalty[i] + 1;
462             }
463         } else if (j == 250 && k == 3) {
464             if (!data.guru[indvPengawas[i][j]) -
465 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
466 1].kodeMapel)) {
467                 penalty[i] = penalty[i] + 1;
468             }
469         } else if (j == 251 && k == 18) {
470             if (!data.guru[indvPengawas[i][j]) -
471 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
472 1].kodeMapel)) {
473                 penalty[i] = penalty[i] + 1;
474             }
475         } else if (j == 252 && k == 32) {
476             if (!data.guru[indvPengawas[i][j]) -
477 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
478 1].kodeMapel)) {
479                 penalty[i] = penalty[i] + 1;
480             }
481         } else if ((j > 252 && j <= 258) && k == 48) {
482             if (!data.guru[indvPengawas[i][j]) -
483 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
484 1].kodeMapel)) {
485                 penalty[i] = penalty[i] + 1;
486             }
487         } else if ((j > 258 && j <= 264) && k == 62) {
488             if (!data.guru[indvPengawas[i][j]) -
489 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
490 1].kodeMapel)) {
491                 penalty[i] = penalty[i] + 1;
492             }
493         } else if ((j > 264 && j <= 270) && k == 75) {
494             if (!data.guru[indvPengawas[i][j]) -
495 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
496 1].kodeMapel)) {
497                 penalty[i] = penalty[i] + 1;
498             }
499         } else if ((j > 270 && j <= 273) && k == 89) {
500             if (!data.guru[indvPengawas[i][j]) -
501 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
502 1].kodeMapel)) {

```

```

503         penalty[i] = penalty[i] + 1;
504     }
505     } else if ((j > 273 && j <= 276) && k == 104) {
506         if (!data.guru[indvPengawas[i][j]) -
507 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
508 1].kodeMapel)) {
509             penalty[i] = penalty[i] + 1;
510         }
511     } else if ((j > 276 && j <= 279) && k == 118) {
512         if (!data.guru[indvPengawas[i][j]) -
513 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
514 1].kodeMapel)) {
515             penalty[i] = penalty[i] + 1;
516         }
517     //HARI 3
518     } else if (j == 280 && k == 5) {
519         if (!data.guru[indvPengawas[i][j]) -
520 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
521 1].kodeMapel)) {
522             penalty[i] = penalty[i] + 1;
523         }
524     } else if (j == 281 && k == 20) {
525         if (!data.guru[indvPengawas[i][j]) -
526 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
527 1].kodeMapel)) {
528             penalty[i] = penalty[i] + 1;
529         }
530     } else if (j == 282 && k == 34) {
531         if (!data.guru[indvPengawas[i][j]) -
532 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
533 1].kodeMapel)) {
534             penalty[i] = penalty[i] + 1;
535         }
536     } else if ((j > 282 && j <= 288) && k == 50) {
537         if (!data.guru[indvPengawas[i][j]) -
538 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
539 1].kodeMapel)) {
540             penalty[i] = penalty[i] + 1;
541         }
542     } else if ((j > 288 && j <= 294) && k == 64) {
543         if (!data.guru[indvPengawas[i][j]) -
544 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
545 1].kodeMapel)) {
546             penalty[i] = penalty[i] + 1;
547         }
548     } else if ((j > 294 && j <= 300) && k == 77) {
549         if (!data.guru[indvPengawas[i][j]) -
550 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
551 1].kodeMapel)) {
552             penalty[i] = penalty[i] + 1;
553         }
554     } else if ((j > 300 && j <= 303) && k == 91) {
555         if (!data.guru[indvPengawas[i][j]) -
556 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
557 1].kodeMapel)) {
558             penalty[i] = penalty[i] + 1;
559         }
560     } else if ((j > 303 && j <= 306) && k == 106) {
561         if (!data.guru[indvPengawas[i][j]) -
562 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
563 1].kodeMapel)) {
564             penalty[i] = penalty[i] + 1;
565         }
566     } else if ((j > 306 && j <= 309) && k == 120) {
567

```

```

578         if (!data.guru[indvPengawas[i][j]) -
579 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
580 1].kodeMapel)) {
581             penalty[i] = penalty[i] + 1;
582         }
583     } else if (j == 310 && k == 7) {
584         if (!data.guru[indvPengawas[i][j]) -
585 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
586 1].kodeMapel)) {
587             penalty[i] = penalty[i] + 1;
588         }
589     } else if (j == 311 && k == 22) {
590         if (!data.guru[indvPengawas[i][j]) -
591 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
592 1].kodeMapel)) {
593             penalty[i] = penalty[i] + 1;
594         }
595     } else if (j == 312 && k == 36) {
596         if (!data.guru[indvPengawas[i][j]) -
597 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
598 1].kodeMapel)) {
599             penalty[i] = penalty[i] + 1;
600         }
601     } else if ((j > 312 && j <= 318) && k == 52) {
602         if (!data.guru[indvPengawas[i][j]) -
603 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
604 1].kodeMapel)) {
605             penalty[i] = penalty[i] + 1;
606         }
607     } else if ((j > 318 && j <= 324) && k == 66) {
608         if (!data.guru[indvPengawas[i][j]) -
609 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
610 1].kodeMapel)) {
611             penalty[i] = penalty[i] + 1;
612         }
613     } else if ((j > 324 && j <= 330) && k == 79) {
614         if (!data.guru[indvPengawas[i][j]) -
615 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
616 1].kodeMapel)) {
617             penalty[i] = penalty[i] + 1;
618         }
619     } else if ((j > 330 && j <= 333) && k == 93) {
620         if (!data.guru[indvPengawas[i][j]) -
621 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
622 1].kodeMapel)) {
623             penalty[i] = penalty[i] + 1;
624         }
625     } else if ((j > 333 && j <= 336) && k == 108) {
626         if (!data.guru[indvPengawas[i][j]) -
627 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
628 1].kodeMapel)) {
629             penalty[i] = penalty[i] + 1;
630         }
631     } else if ((j > 336 && j <= 339) && k == 122) {
632         if (!data.guru[indvPengawas[i][j]) -
633 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
634 1].kodeMapel)) {
635             penalty[i] = penalty[i] + 1;
636         }
637     } else if (j == 340 && k == 9) {
638         if (!data.guru[indvPengawas[i][j]) -
639 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
640 1].kodeMapel)) {
641             penalty[i] = penalty[i] + 1;
642         }

```

```

642         } else if (j == 341 && k == 24) {
643             if (!data.guru[indvPengawas[i][j] -
644 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
645 1].kodeMapel)) {
646                 penalty[i] = penalty[i] + 1;
647             }
648         } else if (j == 342 && k == 38) {
649             if (!data.guru[indvPengawas[i][j] -
650 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
651 1].kodeMapel)) {
652                 penalty[i] = penalty[i] + 1;
653             }
654         } else if ((j > 342 && j <= 348) && k == 54) {
655             if (!data.guru[indvPengawas[i][j] -
656 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
657 1].kodeMapel)) {
658                 penalty[i] = penalty[i] + 1;
659             }
660         } else if ((j > 348 && j <= 354) && k == 68) {
661             if (!data.guru[indvPengawas[i][j] -
662 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
663 1].kodeMapel)) {
664                 penalty[i] = penalty[i] + 1;
665             }
666         } else if ((j > 354 && j <= 360) && k == 81) {
667             if (!data.guru[indvPengawas[i][j] -
668 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
669 1].kodeMapel)) {
670                 penalty[i] = penalty[i] + 1;
671             }
672         } else if ((j > 360 && j <= 363) && k == 95) {
673             if (!data.guru[indvPengawas[i][j] -
674 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
675 1].kodeMapel)) {
676                 penalty[i] = penalty[i] + 1;
677             }
678         } else if ((j > 363 && j <= 366) && k == 110) {
679             if (!data.guru[indvPengawas[i][j] -
680 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
681 1].kodeMapel)) {
682                 penalty[i] = penalty[i] + 1;
683             }
684         } else if ((j > 366 && j <= 369) && k == 124) {
685             if (!data.guru[indvPengawas[i][j] -
686 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
687 1].kodeMapel)) {
688                 penalty[i] = penalty[i] + 1;
689             }
690         } else if (j == 370 && k == 11) {
691             if (!data.guru[indvPengawas[i][j] -
692 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
693 1].kodeMapel)) {
694                 penalty[i] = penalty[i] + 1;
695             }
696         } else if (j == 371 && k == 26) {
697             if (!data.guru[indvPengawas[i][j] -
698 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
699 1].kodeMapel)) {
700                 penalty[i] = penalty[i] + 1;
701             }
702         } else if (j == 372 && k == 42) {
703             if (!data.guru[indvPengawas[i][j] -
704 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k] -
705 1].kodeMapel)) {
706                 penalty[i] = penalty[i] + 1;

```

```

707         }
708         } else if ((j > 372 && j <= 378) && k == 56) {
709             if (!data.guru[indvPengawas[i][j]) -
710 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
711 1].kodeMapel)) {
712                 penalty[i] = penalty[i] + 1;
713             }
714         } else if ((j > 378 && j <= 384) && k == 70) {
715             if (!data.guru[indvPengawas[i][j]) -
716 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
717 1].kodeMapel)) {
718                 penalty[i] = penalty[i] + 1;
719             }
720         } else if ((j > 384 && j <= 390) && k == 83) {
721             if (!data.guru[indvPengawas[i][j]) -
722 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
723 1].kodeMapel)) {
724                 penalty[i] = penalty[i] + 1;
725             }
726         } else if ((j > 390 && j <= 393) && k == 97) {
727             if (!data.guru[indvPengawas[i][j]) -
728 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
729 1].kodeMapel)) {
730                 penalty[i] = penalty[i] + 1;
731             }
732         } else if ((j > 393 && j <= 396) && k == 112) {
733             if (!data.guru[indvPengawas[i][j]) -
734 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
735 1].kodeMapel)) {
736                 penalty[i] = penalty[i] + 1;
737             }
738         } else if ((j > 396 && j <= 399) && k == 126) {
739             if (!data.guru[indvPengawas[i][j]) -
740 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
741 1].kodeMapel)) {
742                 penalty[i] = penalty[i] + 1;
743             }
744         } else if (j == 400 && k == 13) {
745             if (!data.guru[indvPengawas[i][j]) -
746 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
747 1].kodeMapel)) {
748                 penalty[i] = penalty[i] + 1;
749             }
750         } else if (j == 401 && k == 28) {
751             if (!data.guru[indvPengawas[i][j]) -
752 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
753 1].kodeMapel)) {
754                 penalty[i] = penalty[i] + 1;
755             }
756         } else if (j == 402 && k == 44) {
757             if (!data.guru[indvPengawas[i][j]) -
758 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
759 1].kodeMapel)) {
760                 penalty[i] = penalty[i] + 1;
761             }
762         } else if ((j > 402 && j <= 408) && k == 58) {
763             if (!data.guru[indvPengawas[i][j]) -
764 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
765 1].kodeMapel)) {
766                 penalty[i] = penalty[i] + 1;
767             }
768         } else if ((j > 408 && j <= 414) && k == 72) {
769             if (!data.guru[indvPengawas[i][j]) -
770 1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]) -
771 1].kodeMapel)) {

```

772	penalty[i] = penalty[i] + 1;
773	}
774	} else if ((j > 414 && j <= 420) && k == 85) {
775	if (!data.guru[indvPengawas[i][j])
776	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
778	1].kodeMapel)) {
779	penalty[i] = penalty[i] + 1;
780	}
781	} else if ((j > 420 && j <= 423) && k == 99) {
782	if (!data.guru[indvPengawas[i][j])
783	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
784	1].kodeMapel)) {
785	penalty[i] = penalty[i] + 1;
786	}
787	} else if ((j > 423 && j <= 426) && k == 114) {
788	if (!data.guru[indvPengawas[i][j])
789	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
790	1].kodeMapel)) {
791	penalty[i] = penalty[i] + 1;
792	}
793	} else if ((j > 426) && k == 128) {
794	if (!data.guru[indvPengawas[i][j])
795	1].keminatan.equalsIgnoreCase(data.mapel[individu[i][k]
796	1].kodeMapel)) {
797	penalty[i] = penalty[i] + 1;
798	}
799	}
800	penalty2[i] = penalty[i] * 0.25;
801	}
802	}
803	}
804	return penalty2;
805	}
806	
807	

3. Constraint 3

Constraint atau batasan ketiga untuk penyusunan jadwal pengawas adalah apakah terdapat pengawas atau guru senior yang menjaga pada sesi dua. Proses pengecekan *constraint* ini dilakukan dengan mengecek apakah terdapat guru senior yang menjaga pada sesi dua yang ditunjukkan pada Tabel 5.15.

Tabel 5.14 Implementasi kode program constraint pengawas 3

1	public double[] ConstraintPengawas3(int[][] individu) {
2	Data data = new Data();
3	double[] penalty3 = new double[individu.length];
4	double[] penalty = new double[individu.length];
5	for (int i = 0; i < individu.length; i++) {
6	for (int j = 0; j < individu[0].length; j++) {
7	if (j > 219 && j < 430 && data.guru[individu[i][j]
8	- 1].keterangan.equalsIgnoreCase("Senior")) {
9	penalty[i] = penalty[i] + 1;
10	}
11	}
12	penalty3[i] = penalty[i] * 0.25;
13	}
14	return penalty3;
	}

5.2.5 Implementasi Perhitungan Fitness

Proses perhitungan *fitness* dilakukan setelah proses pengecekan *constraint* untuk masing-masing populasi selesai dilakukan. Implementasi perhitungan *fitness* ditunjukkan pada Tabel 5.16.

Tabel 5.15 Implementasi kode program perhitungan *fitness*

```

1 public double[] fitnessMapel(int[][] individu) {
2     GeneticAlgorithm ga = new GeneticAlgorithm();
3     int popSize = individu.length;
4     double[] fitness = new double[popSize];
5     double[] penalty1 = ga.Constraint1(individu);
6     double[] penalty2 = ga.Constraint2(individu);
7     double[] penalty3 = ga.Constraint3(individu);
8     double[] penalty4 = ga.Constraint4(individu);
9     double[] penalty5 = ga.Constraint5(individu);
10    double fitnessTotal = 0;
11    for (int i = 0; i < popSize; i++) {
12        fitnessTotal = penalty1[i] + penalty2[i] + penalty3[i]
13        + penalty4[i] + penalty5[i];
14        fitness[i] = 1 / (1 + fitnessTotal);
15        System.out.println("P" + (i + 1) + ": " + fitness[i]);
16    }
17    return fitness;
18 }

```

5.2.6 Implementasi Seleksi

Proses seleksi yang digunakan pada penelitian ini adalah elitism selection. Metode seleksi ini akan memilih individu generasi selanjutnya dengan mengurutkan *fitness* dari yang paling baik. Implementasi metode elitism selection ini ditunjukkan pada Tabel 5.17.

Tabel 5.16 Implementasi kode program *elitism selection*

```

1 public int[][] ElitismSelection(int[][] individu, int popSize) {
2     GeneticAlgorithm ga = new GeneticAlgorithm();
3     int[][] hasilSeleksi = new
4     int[popSize][individu[0].length];
5     double[] fitness = ga.fitnessMapel(individu);
6     int length = individu.length;
7     for (int i = 0; i < length - 1; i++) {
8         double temp;
9         int indexTemp = 0;
10        for (int j = 0; j < (length - i - 1); j++) {
11            if (fitness[j] < fitness[j + 1]) {
12                temp = fitness[j];
13                fitness[j] = fitness[j + 1];
14                fitness[j + 1] = temp;
15                for (int k = 0; k < individu[0].length; k++) {
16                    indexTemp = individu[i][k];
17                    individu[j][k] = individu[j + 1][k];
18                    individu[j + 1][k] = indexTemp;
19                }
20            }
21        }
22    }
23    for (int i = 0; i < popSize; i++) {
24        for (int j = 0; j < individu[0].length; j++) {
25            hasilSeleksi[i][j] = individu[i][j];
26        }
27    }
28    return hasilSeleksi;
29 }

```



```

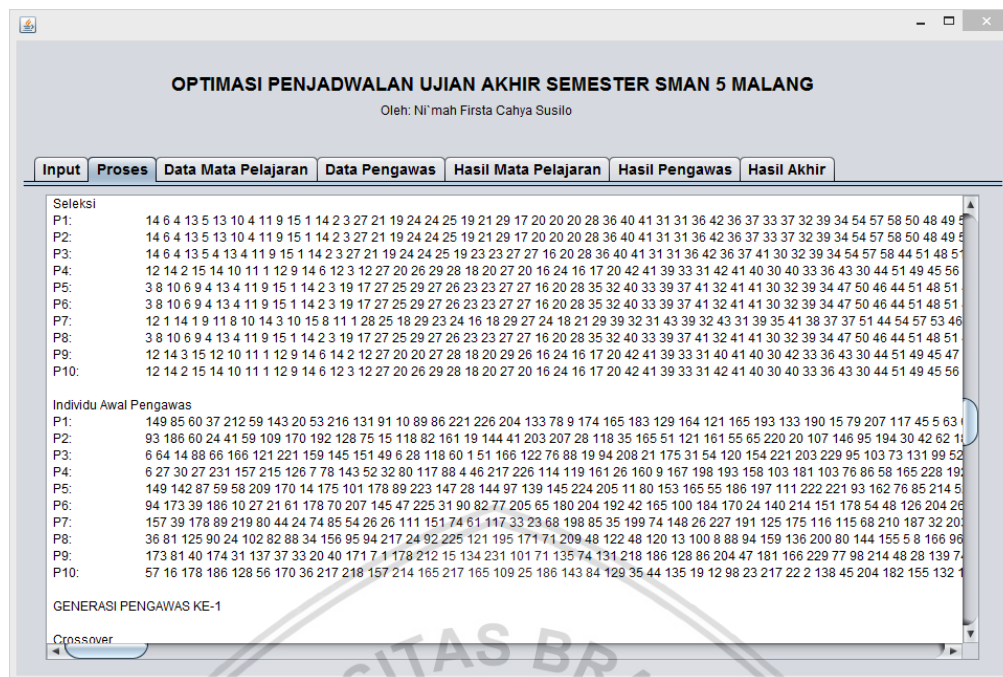
26         }
27     }
28     for (int i = 0; i < popSize; i++) {
29         System.out.println(fitness[i]);
30     }
31     return hasilSeleksi;
32 }

```

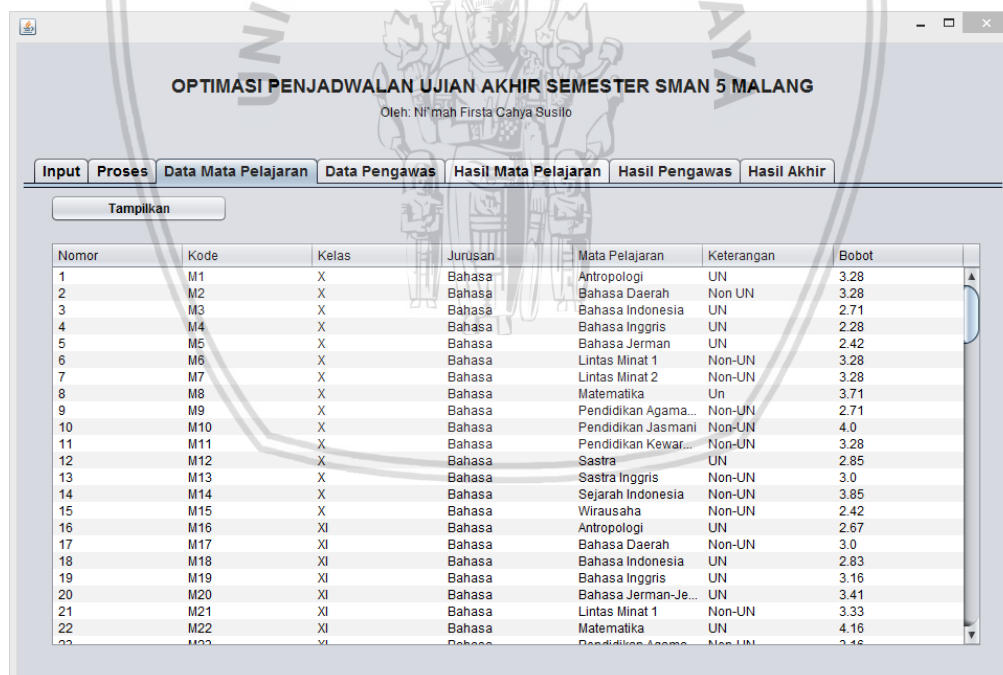
5.3 Implementasi Antar Muka

Implementasi sistem penjadwalan ujian akhir semester pada penelitian ini terdiri dari tujuh halaman. Pada halaman input berfungsi untuk menerima inputan berupa ukuran populasi, jumlah kombinasi serta nilai *crossover rate* dan *mutation rate* yang ditunjukkan pada Gambar 5.1 dan pada Gambar 5.2 ditunjukkan proses perhitungan dengan algoritme genetika. Pada Gambar 5.3 dan Gambar 5.4 ditunjukkan data mata pelajaran dan data pengawas yang digunakan dalam penelitian. Individu awal serta individu akhir untuk masing-masing mata pelajaran dan pengawas ditunjukkan pada Gambar 5.5 dan Gambar 5.6. Hasil penjadwalan mata pelajaran dan pengawas ditunjukkan pada Gambar 5.7 dan Gambar 5.8.

Gambar 5.1 Antarmuka halaman input



Gambar 5.2 Antarmuka halaman proses



Gambar 5.3 Halaman data mata pelajaran

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER SMAN 5 MALANG
Oleh: Ni'mah Firsta Cahya Susilo

Input **Proses** **Data Mata Pelajaran** **Data Pengawas** **Hasil Mata Pelajaran** **Hasil Pengawas** **Hasil Akhir**

Tampilkan

Nomor	Kode	Nama	Keminatan	Keterangan
98	G29	Drs. Suromargono	M11	Senior
99	G29	Drs. Suromargono	M84	Senior
100	G30	Drs. Tamtama Rahardja	M7	Senior
101	G30	Drs. Tamtama Rahardja	M93	Senior
102	G30	Drs. Tamtama Rahardja	M50	Senior
103	G30	Drs. Tamtama Rahardja	M64	Senior
104	G31	Drs. Tavip Ds	M105	Senior
105	G31	Drs. Tavip Ds	M90	Senior
106	G31	Drs. Tavip Ds	M86	Senior
107	G31	Drs. Tavip Ds	M129	Senior
108	G32	Drs. Tjatur Imbang S. M.Pd	M93	Senior
109	G32	Drs. Tjatur Imbang S. M.Pd	M7	Senior
110	G32	Drs. Tjatur Imbang S. M.Pd	M21	Senior
111	G32	Drs. Tjatur Imbang S. M.Pd	M107	Senior
112	G32	Drs. Tjatur Imbang S. M.Pd	M76	Senior
113	G32	Drs. Tjatur Imbang S. M.Pd	M48	Senior
114	G32	Drs. Tjatur Imbang S. M.Pd	M62	Senior
115	G33	Dyah Ayu Puspitasari, M.Pd	M94	Junior
116	G34	Eko Prasetyo, S.Pai	M123	Junior
117	G34	Eko Prasetyo, S.Pai	M109	Junior
118	G34	Eko Prasetyo, S.Pai	M68	Junior
119	G35	Elfata, S.Pd	M45	Junior
120	G35	Elfata, S.Pd	M12	Junior

Gambar 5.4 Halaman data pengawas

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER SMAN 5 MALANG
Oleh: Ni'mah Firsta Cahya Susilo

Input **Proses** **Data Mata Pelajaran** **Data Pengawas** **Hasil Mata Pelajaran** **Hasil Pengawas** **Hasil Akhir**

Individu Awal Mata Pelajaran

Individu	Kromosom
1	[15, 13, 5, 14, 4, 8, 1, 13, 4, 2, 15, 8, 12, 5, 13, 26, 21, 18, 16, 23, 26, 24, 27, 18, ...
2	[14, 6, 4, 13, 5, 13, 10, 12, 1, 3, 14, 9, 10, 12, 10, 27, 21, 19, 24, 24, 25, 19, 21, 2, ...
3	[12, 14, 2, 15, 12, 10, 11, 1, 12, 9, 9, 8, 5, 6, 7, 27, 20, 26, 27, 28, 18, 20, 29, 20, ...
4	[14, 1, 12, 14, 2, 4, 14, 1, 11, 5, 7, 10, 8, 12, 4, 28, 21, 22, 26, 23, 29, 19, 18, 21, ...
5	[12, 1, 14, 1, 9, 11, 8, 10, 15, 3, 14, 10, 8, 11, 1, 28, 23, 18, 29, 25, 24, 16, 18, 29, ...
6	[8, 6, 4, 7, 15, 12, 2, 4, 7, 3, 14, 6, 14, 3, 12, 23, 27, 22, 28, 20, 19, 17, 27, 18, 16, ...
7	[5, 9, 8, 7, 1, 7, 7, 1, 4, 7, 12, 11, 1, 9, 8, 25, 24, 29, 29, 28, 23, 28, 28, 22, 16, 17, ...
8	[9, 12, 1, 8, 13, 8, 7, 11, 12, 9, 15, 12, 13, 10, 1, 16, 29, 16, 27, 26, 17, 28, 21, 25, ...
9	[3, 8, 10, 6, 9, 4, 13, 4, 11, 9, 15, 1, 14, 2, 3, 19, 17, 27, 25, 29, 27, 26, 23, 23, 27, ...
10	[10, 14, 14, 1, 15, 12, 1, 6, 9, 3, 13, 8, 5, 10, 8, 21, 28, 16, 25, 22, 22, 28, 19, 21, ...

Individu Akhir Mata Pelajaran

Individu	Kromosom	Fitness
1	[14, 6, 4, 13, 5, 13, 10, 4, 11, 9, 15, 1, 14, 2, 3, 27, 21, ...	0.0769941484447182
2	[14, 6, 4, 13, 5, 13, 10, 4, 11, 9, 15, 1, 14, 2, 3, 27, 21, ...	0.0769941484447182
3	[14, 6, 4, 13, 5, 4, 13, 4, 11, 9, 15, 1, 14, 2, 3, 27, 21, ...	0.07506943923128895
4	[12, 14, 2, 15, 14, 10, 11, 1, 12, 9, 14, 6, 12, 3, 12, 27, ...	0.06876633200385092
5	[3, 8, 10, 6, 9, 4, 13, 4, 11, 9, 15, 1, 14, 2, 3, 19, 17, 2, ...	0.07037792948131467
6	[3, 8, 10, 6, 9, 4, 13, 4, 11, 9, 15, 1, 14, 2, 3, 19, 17, 2, ...	0.07037792948131467
7	[12, 1, 14, 1, 9, 11, 8, 10, 14, 3, 10, 15, 8, 11, 1, 28, 2, ...	0.07037792948131467
8	[3, 8, 10, 6, 9, 4, 13, 4, 11, 9, 15, 1, 14, 2, 3, 19, 17, 2, ...	0.07037792948131467
9	[12, 14, 3, 15, 12, 10, 11, 1, 12, 9, 14, 6, 14, 2, 12, 27, ...	0.06983240223463688
10	[12, 14, 2, 15, 14, 10, 11, 1, 12, 9, 14, 6, 12, 3, 12, 27, ...	0.06876633200385092

Gambar 5.5 Halaman hasil mata pelajaran

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER SMAN 5 MALANG
Oleh: Ni'mah Firsta Cahya Susilo

Input **Proses** **Data Mata Pelajaran** **Data Pengawas** **Hasil Mata Pelajaran** **Hasil Pengawas** **Hasil Akhir**

Individu Awal Pengawas

Individu	Kromosom
1	[149, 85, 60, 37, 212, 59, 143, 20, 53, 216, 131, 91, 10, 89, 86, 221, 226, 204, 1...
2	[93, 186, 60, 24, 41, 59, 109, 170, 192, 128, 75, 15, 118, 82, 161, 19, 144, 41, 2...
3	[6, 64, 14, 88, 66, 166, 121, 221, 159, 145, 151, 49, 6, 28, 118, 60, 1, 51, 166, 12...
4	[6, 27, 30, 27, 231, 157, 215, 126, 7, 78, 143, 52, 32, 80, 117, 88, 4, 46, 217, 226...
5	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89, 223, 147, 28, 144, 97, 13...
6	[94, 173, 39, 186, 10, 27, 21, 61, 178, 70, 207, 145, 47, 225, 31, 90, 82, 77, 205...
7	[157, 39, 178, 89, 219, 80, 44, 24, 74, 85, 54, 26, 26, 111, 151, 74, 61, 117, 33, 2...
8	[36, 81, 125, 90, 24, 102, 82, 88, 34, 156, 95, 94, 217, 24, 92, 225, 121, 195, 17...
9	[173, 81, 40, 174, 31, 137, 37, 33, 20, 40, 171, 7, 1, 178, 212, 15, 134, 231, 101...
10	[57, 16, 178, 186, 128, 56, 170, 36, 217, 218, 157, 214, 165, 217, 165, 109, 25...

Individu Akhir Pengawas

Individu	Kromosom	Fitness
1	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.001989060169070114
2	[6, 64, 14, 88, 66, 166, 121, 221, 159, 145, 151, 49, 6...	0.0020050125313283208
3	[36, 81, 125, 90, 24, 102, 82, 88, 34, 156, 95, 94, 217...	0.0018770530267480056
4	[36, 81, 125, 90, 24, 102, 82, 88, 34, 156, 95, 94, 217...	0.001876172607879925
5	[6, 64, 14, 88, 66, 166, 121, 221, 159, 145, 151, 49, 6...	0.0020202020202020202
6	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.0020161290322580645
7	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.0020202020202020202
8	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.0019474196689386564
9	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.001948368241597662
10	[149, 142, 87, 59, 58, 209, 170, 14, 175, 101, 178, 89...	0.0020232675771370764

Gambar 5.6 Halaman hasil pengawas

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER SMAN 5 MALANG
Oleh: Ni'mah Firsta Cahya Susilo

Input **Proses** **Data Mata Pelajaran** **Data Pengawas** **Hasil Mata Pelajaran** **Hasil Pengawas** **Hasil Akhir**

Jadwal Mata Pelajaran **Jadwal Pengawas**

Hari	Jam	Kelas	Mata Pelajaran
Senin	07.00 - 08.30	X Bahasa	Sejarah Indonesia
Senin	08.30 - 09.30	X Bahasa	Lintas Minat 1
Selasa	07.00 - 08.30	X Bahasa	Bahasa Inggris
Selasa	08.30 - 09.30	X Bahasa	Sastra Inggris
Rabu	07.00 - 08.30	X Bahasa	Bahasa Jerman
Rabu	08.30 - 09.30	X Bahasa	Sastra Inggris
Kamis	07.00 - 08.30	X Bahasa	Pendidikan Jasmani
Kamis	08.30 - 09.30	X Bahasa	Bahasa Inggris
Jum'at	07.00 - 08.30	X Bahasa	Pendidikan Kewarganegaraan
Jum'at	08.30 - 09.30	X Bahasa	Pendidikan Agama Islam
Sabtu	07.00 - 08.30	X Bahasa	Wirausaha
Sabtu	08.30 - 09.30	X Bahasa	Antropologi
Senin	07.00 - 08.30	X Bahasa	Sejarah Indonesia
Senin	08.30 - 09.30	X Bahasa	Bahasa Daerah
Selasa	07.00 - 08.30	X Bahasa	Bahasa Indonesia
Senin	07.00 - 08.30	XI Bahasa	Sastra
Senin	08.30 - 09.30	XI Bahasa	Lintas Minat 1
Selasa	07.00 - 08.30	XI Bahasa	Bahasa Inggris
Selasa	08.30 - 09.30	XI Bahasa	Pendidikan Jasmani
Rabu	07.00 - 08.30	XI Bahasa	Pendidikan Kewarganegaraan
Rabu	08.30 - 09.30	XI Bahasa	Bahasa Inggris
Kamis	07.00 - 08.30	XI Bahasa	Lintas Minat 1
Kamis	08.30 - 09.30	XI Bahasa	Wirausaha
Jum'at	07.00 - 08.30	XI Bahasa	

Gambar 5.7 Halaman hasil jadwal mata pelajaran

OPTIMASI PENJADWALAN UJIAN AKHIR SEMESTER SMAN 5 MALANG				
Oleh: Ni'mah Firsta Cahya Susilo				
Input	Proses	Data Mata Pelajaran	Data Pengawas	Hasil Mata Pelajaran
Hasil Pengawas	Hasil Akhir			
Jadwal Mata Pelajaran	Jadwal Pengawas			
Hari	Jam	Kelas	Mata Pelajaran	Pengawas
Jum'at	07.00 - 08.30	XI IPS	Bahasa Inggris	Dra. Yayuk Ernawati
Jum'at	07.00 - 08.30	XI IPS	Bahasa Inggris	Dra. Ruhaningsih H.
Jum'at	07.00 - 08.30	XII IPS	Matematika	Dra. Heri Rini
Jum'at	07.00 - 08.30	XII IPS	Matematika	Laily Fitria Wati, S.Pd., Gr
Jum'at	07.00 - 08.30	XII IPS	Matematika	Drs. Suromargono
Sabtu	07.00 - 08.30	X Bahasa	Wirausaha	Suyoso, M.Pd
Sabtu	07.00 - 08.30	XI Bahasa	Bahasa Jerman-Jepang	Sulis Hanifah, S.Pd
Sabtu	07.00 - 08.30	XII Bahasa	Pendidikan Agama Islam	Rofida Murniah
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Agustina Puji Astuti
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Titik Wulandari
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Elfata, S.Pd
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Rizqi Ariska Rama
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Rizqi Ariska Rama
Sabtu	07.00 - 08.30	X MIPA	Lintas Minat 2	Wahyu Esti Rahmawati, S.Pd
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Agung Setyawanto, S.Pd
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Dra. Sri Setyawati
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Mukhamad Hidayatullah, S.Ps
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Irma Aryan, S.Pd
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Agung Setyawanto, S.Pd
Sabtu	07.00 - 08.30	XI MIPA	Pendidikan Agama Islam	Dra. Retno Sutriani
Sabtu	07.00 - 08.30	XII MIPA	Biologi	Chandra Nuri M, S.Pd
Sabtu	07.00 - 08.30	XII MIPA	Biologi	Elfata, S.Pd
Sabtu	07.00 - 08.30	XII MIPA	Biologi	Dedi Pambela, S.Pd
Sabtu	07.00 - 08.30	XII MIPA	Biologi	Drs. Djoko Budianto
Sabtu	07.00 - 08.30	XII MIPA	Biologi	Drs. Suwanto

Gambar 5.8 Halaman hasil jadwal pengawas

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian pada sistem penjadwalan ujian akhir semester menggunakan algoritme genetika ini dilakukan terhadap parameter kombinasi nilai *crossover rate* dan *mutation rate*, ukuran populasi serta jumlah generasi. Hal ini bertujuan untuk mengetahui bagaimana pengaruh parameter-parameter tersebut terhadap nilai *fitness* sehingga dapat menghasilkan solusi penjadwalan yang optimal dengan minimalnya nilai pelanggaran. Pengujian ini dilakukan sebanyak 5 kali terhadap masing-masing pengujian yang akan dihitung rata-rata nilai *fitness* yang telah dihasilkan.

6.1 Hasil Pengujian Terhadap Ukuran Populasi

Pengujian yang pertama adalah pengujian terhadap ukuran populasi untuk menemukan ukuran populasi yang terbaik. Jumlah populasi yang diuji yaitu sebesar 10 hingga 200 populasi dengan generasi sebesar 500 dan kombinasi nilai *crossover rate* dan *mutation rate* masing-masing sebesar 0.5 yang dilakukan pada kedua individu mata pelajaran dan kromosom.

Berdasarkan hasil pengujian pada Tabel 6.1 ukuran populasi mengalami peningkatan dari ukuran populasi terkecil yaitu 10 populasi. Pada ukuran populasi 20 sampai 60 rata-rata nilai *fitness* mengalami peningkatan. Tetapi pada ukuran populasi 70 nilai rata-rata *fitness* mengalami penurunan dan pada ukuran populasi 80 rata-rata nilai *fitness* mengalami peningkatan kembali. Hal ini disebabkan area pencarian pada populasi telah semakin luas hingga ditemukan nilai rata-rata *fitness* terbesar pada percobaan ditemukan pada ukuran populasi 180 yang terlihat pada Gambar 6.1. Rata-rata nilai *fitness* terbesar adalah 0.144613584 ditemukan pada ukuran populasi 180 pada bagian tabel yang berwarna kuning dengan waktu komputasi sebesar 30,037 detik. Sedangkan untuk nilai *fitness* terkecil ditemukan pada ukuran populasi 10 dengan nilai rata-rata *fitness* sebesar 0.1005 dan waktu komputasi 1440.62 detik. Rata-rata nilai *fitness* terkecil ditemukan pada populasi 10 yang disebabkan oleh ruang pencarian pada populasi tersebut masih sangat sempit.

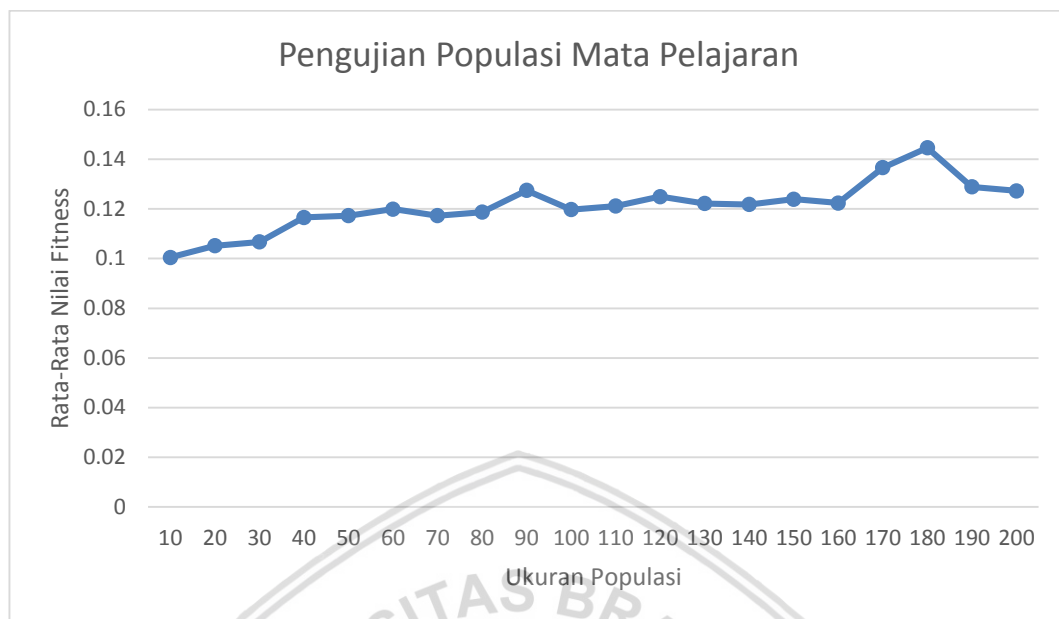
Pada pengujian ukuran populasi pengawas mengalami peningkatan sesuai dengan peningkatan ukuran populasi. Hal ini disebabkan ukuran pencarian populasi yang semakin luas. Berdasarkan Tabel 6.2 rata-rata nilai *fitness* terbesar adalah 0.004503 pada bagian tabel yang berwarna kuning dengan ukuran populasi 150 dengan waktu komputasi sebesar 28769.2 detik. Sedangkan untuk nilai *fitness* terkecil ditemukan pada nilai 0.002146 dengan ukuran populasi 10 dengan waktu komputasi 1780.4 detik. Pada Gambar 6.2 ditunjukkan grafik hasil pengujian ukuran populasi pengawas yang terus mengalami peningkatan hingga pada populasi 150 mencapai rata-rata nilai *fitness* terbesar sejumlah 0.004503 dengan waktu komputasi 28769.2 detik. Nilai rata-rata *fitness* pengawas terkecil ditemukan pada populasi 10 dengan *fitness* sebesar 0.002146 dengan waktu komputasi 1780.4 detik.

Tabel 6.1 Pengujian ukuran populasi pada individu mata pelajaran

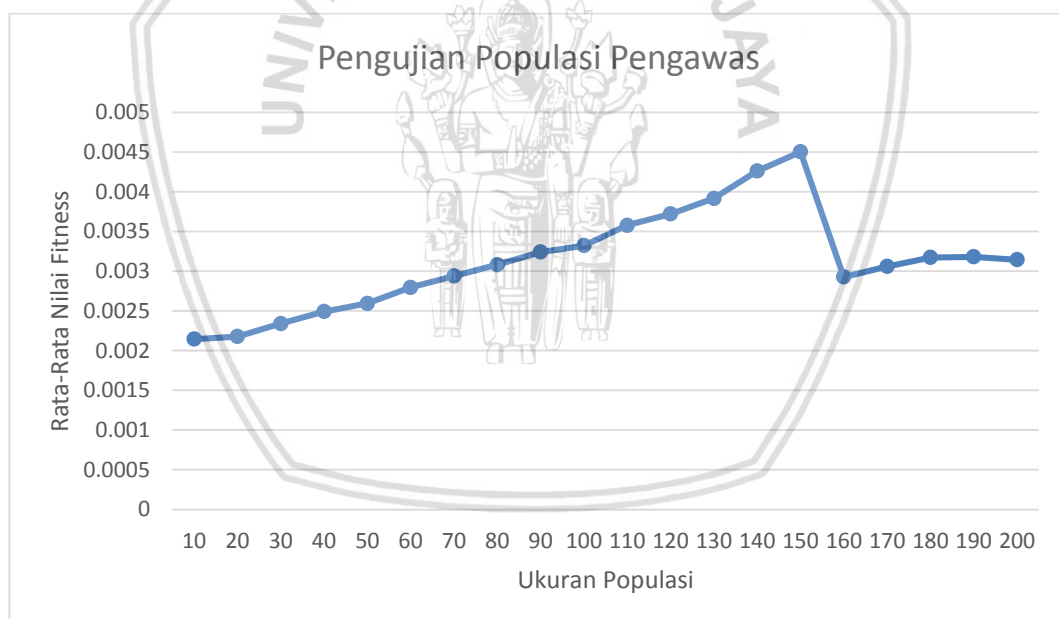
Populasi	Percobaan ke-										Rata-rata	
	1		2		3		4		5			
	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>
10	1710	0.10121	1702	0.09899	1701	0.10121	1699	0.10724	2090	0.09384	1780.4	0.100498619
20	3390	0.11121	3358	0.09899	3389	0.10853	3382	0.10354	3781	0.10354	3460	0.105162594
30	5583	0.10354	5103	0.10474	5576	0.10121	5152	0.11549	5255	0.10853	5333.8	0.10670358
40	7385	0.13055	7386	0.11853	7781	0.12511	7789	0.10985	7285	0.09899	7525.2	0.116605502
50	9049	0.10853	9057	0.11403	9076	0.10236	9035	0.13648	9030	0.12511	9049.4	0.117302237
60	10752	0.11403	11410	0.12011	10758	0.1234	11417	0.11699	10377	0.12511	10942.8	0.119924507
70	12493	0.11699	12469	0.11853	12477	0.12011	12496	0.11549	15804	0.11549	13147.8	0.117318244
80	13624	0.13247	13568	0.11121	13689	0.12173	13714	0.11403	13916	0.11403	13702.2	0.118691311
90	15295	0.11549	15304	0.15803	15686	0.11549	15291	0.12511	15526	0.1234	15420.4	0.127501338
100	14236	0.11853	14245	0.11853	13517	0.12687	13474	0.12511	13479	0.10985	13790.2	0.11977716
110	20594	0.11699	20657	0.13444	20621	0.11403	20698	0.12173	19325	0.11853	20379	0.121142069
120	20424	0.13247	19726	0.11699	20366	0.1234	20476	0.1234	20788	0.12868	20356	0.124985916
130	20788	0.11699	21726	0.12868	20671	0.12511	19887	0.11853	27695	0.12173	22153.4	0.122206711
140	26005	0.11549	23877	0.12011	26005	0.11549	29950	0.13444	23744	0.1234	25916.2	0.121783969
150	28711	0.13055	25964	0.1234	28700	0.11549	34507	0.12687	25964	0.1234	28769.2	0.123939628
160	23899	0.11549	23893	0.11699	25914	0.13444	25924	0.1234	26581	0.12173	25242.2	0.12240847
170	28761	0.15803	29006	0.12173	29451	0.16085	29421	0.12868	32732	0.11403	29874.2	0.136662859
180	34257	0.13648	26056	0.14767	31562	0.14767	25747	0.15267	32563	0.13858	30037	0.144613584
190	35874	0.13247	35842	0.13448	35765	0.1234	34562	0.1234	35805	0.13075	35569.6	0.128897409
200	36746	0.13027	36912	0.12167	37523	0.13144	36573	0.12055	37303	0.13258	37011.4	0.127302641

Tabel 6.2 Pengujian ukuran populasi pada individu pengawas

Populasi	Percobaan ke-										Rata-rata	
	1		2		3		4		5			
	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>
10	1710	0.002159	1702	0.002064	1701	0.002148	1699	0.002131	2090	0.002227	1780.4	0.002146
20	3390	0.002206	3358	0.00216	3389	0.002283	3382	0.002104	3781	0.002128	3460	0.002176
30	5583	0.002298	5103	0.002395	5576	0.002282	5152	0.002346	5255	0.002382	5333.8	0.002341
40	7385	0.002497	7386	0.002524	7781	0.002576	7789	0.002465	7285	0.002397	7525.2	0.002491
50	9049	0.002604	9057	0.002677	9076	0.002538	9035	0.002602	9030	0.002546	9049.4	0.002594
60	10752	0.002679	11410	0.002789	10758	0.002841	11417	0.002805	10377	0.002859	10942.8	0.002795
70	12493	0.002915	12469	0.00303	12477	0.002972	12496	0.00292	15804	0.002855	13147.8	0.002938
80	13624	0.003075	13568	0.003123	13689	0.00314	13714	0.003063	13916	0.003005	13702.2	0.003081
90	15295	0.003306	15304	0.003255	15686	0.003152	15291	0.003142	15526	0.003353	15420.4	0.003242
100	14236	0.003475	14245	0.003265	13517	0.003255	13474	0.003333	13479	0.003276	13790.2	0.003321
110	20594	0.003604	20657	0.003448	20621	0.003594	20698	0.003731	19325	0.003509	20379	0.003577
120	20424	0.00364	19726	0.003839	20366	0.003687	20476	0.003594	20788	0.003846	20356	0.003721
130	20788	0.003549	21726	0.003902	20671	0.004004	19887	0.004057	27695	0.004069	22153.4	0.003916
140	26005	0.004301	23877	0.004357	26005	0.004301	29950	0.004315	23744	0.004032	25916.2	0.004261
150	28711	0.004695	25964	0.00451	28700	0.004324	34507	0.004474	25964	0.00451	28769.2	0.004503
160	23899	0.003017	23893	0.00295	25914	0.003005	25924	0.002851	26581	0.002815	25242.2	0.002928
170	28761	0.003203	29006	0.00306	29451	0.003084	29421	0.002987	32732	0.002965	29874.2	0.00306
180	34257	0.003195	26056	0.003532	31562	0.003063	25747	0.002972	32563	0.003101	30037	0.003172
190	35874	0.003252	35842	0.003115	35765	0.003094	34562	0.003239	35805	0.003208	35569.6	0.003181
200	36746	0.003157	36912	0.003123	37523	0.003147	36573	0.003134	37303	0.003157	37011.4	0.003144



Gambar 6.1 Grafik pengujian ukuran populasi mata pelajaran



Gambar 6.2 Grafik pengujian ukuran populasi pengawas

6.2 Hasil Pengujian Terhadap Kombinasi Crossover rate (Cr) dan Mutation Rate (Mr)

Pengujian yang kedua adalah pengujian terhadap nilai kombinasi *crossover rate* dan *mutation rate* yang bertujuan untuk menemukan kombinasi nilai *crossover rate* dan *mutation rate* yang optimasi. Jumlah individu yang digunakan dalam pengujian ini adalah sebesar 150 populasi yang telah diperoleh pada pengujian sebelumnya sebagai populasi optimal. Jumlah generasi sebesar 500

yang dilakukan sebanyak 5 kali percobaan pada setiap kombinasi Cr dan Mr. Pengujian ini juga dilakukan pada kedua individu mata pelajaran dan kromosom. Hasil pengujian terhadap kombinasi nilai Cr dan Mr pada individu mata pelajaran dan individu pengawas ditunjukkan pada Tabel 6.3 dan 6.4.

Hasil dari pengujian kombinasi nilai Cr dan Mr pada populasi mata pelajaran ditunjukkan pada Gambar 6.3. Berdasarkan grafik tersebut nilai rata-rata *fitness* mengalami peningkatan dari kombinasi Cr = 0.9 dan Mr = 0.1 hingga kombinasi Cr = 0.7 dan Mr = 0.3. Tetapi pada kombinasi Cr = 0.6 dan Mr = 0.4 rata-rata nilai *fitness* mengalami penurunan hingga pada kombinasi Cr = 0.1 dan Mr = 0.9. Nilai rata-rata *fitness* terbesar ditemukan pada populasi dengan kombinasi Cr = 0.7 dan Mr = 0.3 sebesar 0.14083 dengan waktu komputasi 15295 detik pada bagian tabel yang berwarna kuning. Nilai rata-rata *fitness* terkecil ditemukan pada populasi dengan kombinasi Cr = 0.1 dan Mr = 0.1 sebesar 0.10564 dengan waktu komputasi 15451 detik. Nilai kombinasi *crossover rate* dan *mutation rate* optimal yang ditemukan menyebabkan algoritme genetika mengalami penurunan diversitas karena jumlah *offspring crossover* memiliki jumlah yang lebih daripada *offspring* mutasi. *Crossover rate* yang tinggi dari pada *mutation rate* cenderung menyebabkan *offspring* memiliki kemiripan dengan induknya.

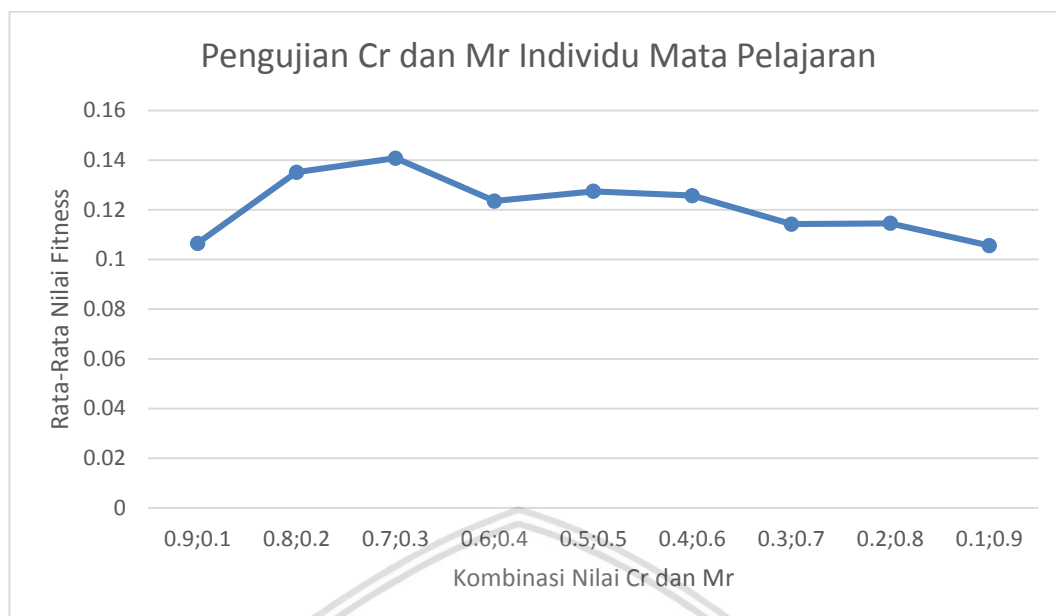
Pada pengujian kombinasi nilai Cr dan Mr pada populasi pengawas mengalami peningkatan dan penurunan berdasarkan nilai kombinasi yang diujikan. Berdasarkan Tabel 6.4 rata-rata nilai *fitness* terbesar adalah 0.00329 dengan nilai *crossover rate* sebesar 0.6 dan nilai *mutation rate* sebesar 0.4 dengan waktu komputasi sebesar 15621.2 detik pada bagian tabel yang berwarna kuning. Sedangkan untuk nilai *fitness* terkecil ditemukan pada nilai 0.00312 dengan *crossover rate* sebesar 0.1 dan *mutation rate* sebesar 0.9 dengan waktu komputasi 15451 detik. Pada Gambar 6.1 ditunjukkan grafik hasil pengujian kombinasi *crossover rate* dan *mutation rate* pada individu mata pelajaran. Nilai kombinasi Cr dan Mr optimal yang ditemukan menyebabkan algoritme genetika kehilangan kesempatan untuk melakukan eksplorasi karena memiliki *offspring crossover* yang lebih daripada *offspring* mutasi.

Tabel 6.3 Hasil pengujian kombinasi Cr dan Mr pada individu mata pelajaran

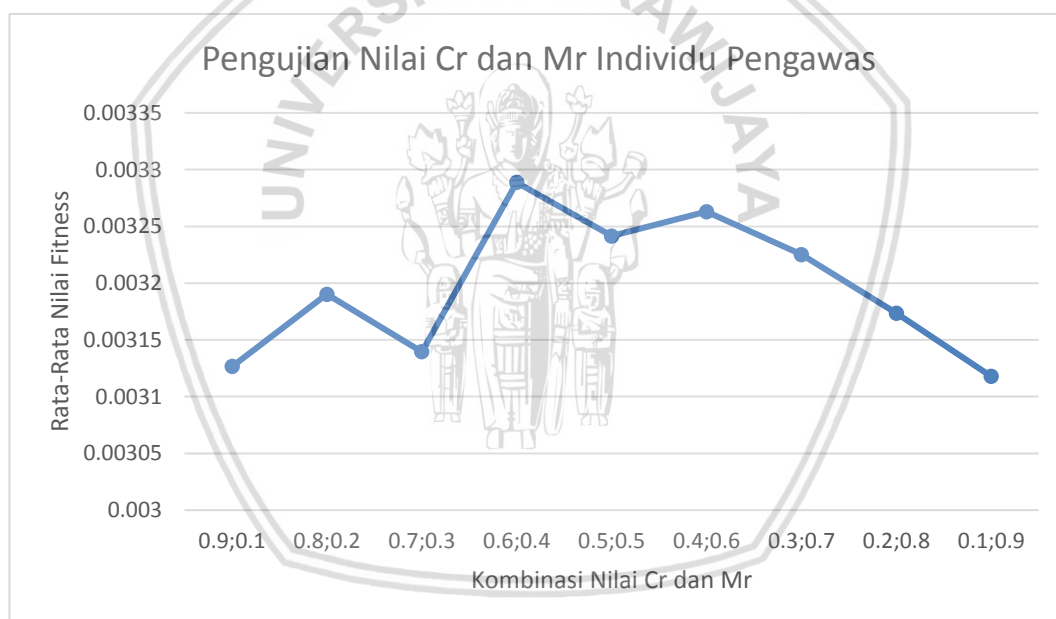
Kombinasi		Percobaan ke-										Rata-rata	
		1		2		3		4		5			
Cr	Mr	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>
0.9	0.1	16603	0.10853	25058	0.11549	18190	0.10236	19907	0.09899	19999	0.10724	19951.4	0.10652
0.8	0.2	18016	0.10236	17693	0.1126	13684	0.15267	15245	0.15803	19116	0.15013	16750.8	0.13516
0.7	0.3	15235	0.12011	15222	0.15267	15351	0.14298	15339	0.14767	15328	0.14075	15295	0.14083
0.6	0.4	15351	0.1234	15728	0.11121	15668	0.15267	15684	0.10853	15675	0.12173	15621.2	0.12351
0.5	0.5	15295	0.11549	15304	0.15803	15686	0.11549	15291	0.12511	15526	0.1234	15420.4	0.1275
0.4	0.6	17286	0.13055	16368	0.12173	17277	0.12011	15700	0.12173	19383	0.13444	17202.8	0.12571
0.3	0.7	18893	0.10354	18602	0.12511	17735	0.11853	19351	0.11549	18679	0.10853	18652	0.11424
0.2	0.8	17562	0.11403	13622	0.12011	13620	0.12173	15896	0.10724	15906	0.10985	15321.2	0.11459
0.1	0.9	15855	0.1126	15723	0.10354	15700	0.10009	17168	0.10598	12809	0.10598	15451	0.10564

Tabel 6.4 Hasil pengujian kombinasi Cr dan Mr pada individu pengawas

Kombinasi		Percobaan ke-										Rata-rata	
		1		2		3		4		5			
Cr	Mr	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>	Waktu (detik)	<i>fitness</i>
0.9	0.1	16603	0.00307	25058	0.00307	18190	0.00318	19907	0.00321	19999	0.0031	19951.4	0.00313
0.8	0.2	18016	0.00302	17693	0.00332	13684	0.00319	15245	0.00322	19116	0.0032	16750.8	0.00319
0.7	0.3	15235	0.00316	15222	0.0031	15351	0.00307	15339	0.00326	15328	0.00311	15295	0.00314
0.6	0.4	15351	0.00338	15728	0.0033	15668	0.00323	15684	0.00329	15675	0.00324	15621.2	0.00329
0.5	0.5	15295	0.00331	15304	0.00325	15686	0.00315	15291	0.00314	15526	0.00335	15420.4	0.00324
0.4	0.6	17286	0.00324	16368	0.00327	17277	0.00328	15700	0.00318	19383	0.00334	17202.8	0.00326
0.3	0.7	18893	0.00316	18602	0.00329	17735	0.00323	19351	0.00327	18679	0.00317	18652	0.00323
0.2	0.8	17562	0.00335	13622	0.00306	13620	0.00314	15896	0.00301	15906	0.0033	15321.2	0.00317
0.1	0.9	15855	0.00299	15723	0.0032	15700	0.00312	17168	0.00316	12809	0.00311	15451	0.00312



Gambar 6.3 Grafik pengujian kombinasi Nilai Cr dan Mr individu mata pelajaran

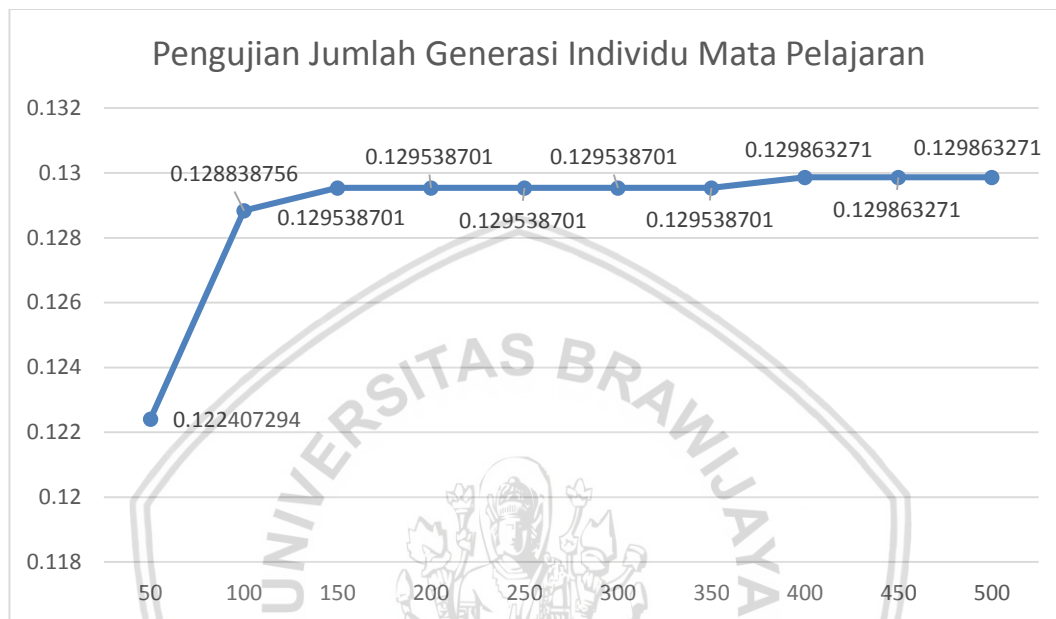


Gambar 6.4 Grafik hasil pengujian kombinasi nilai Cr dan Mr individu pengawas

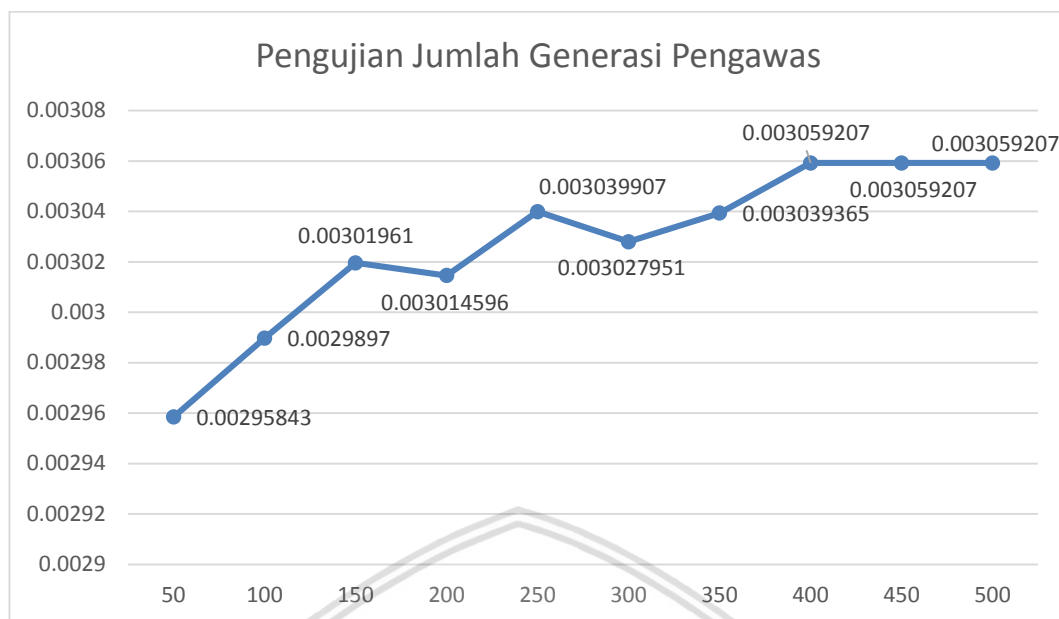
6.3 Hasil Pengujian Terhadap Jumlah Generasi

Pengujian yang selanjutnya adalah pengujian terhadap jumlah generasi. Pada sistem ini pengujian generasi tidak dilakukan secara terpisah seperti dua pengujian sebelumnya. Hasil pengujian terhadap jumlah generasi didapatkan dari hasil pengujian sebelumnya yaitu pengujian kombinasi nilai *crossover rate* dan nilai *mutation rate*. Pengujian dilakukan dengan ukuran populasi optimal 180 dan kombinasi nilai Cr sebesar 0.7 serta Mr sebesar 0.3. Kondisi rata-rata *fitness* mata pelajaran mulai terlihat stabil pada jumlah generasi 150 hingga generasi ke-400 yang terlihat cenderung naik dengan selisih yang tidak signifikan. Hal ini

ditunjukkan pada Gambar 6.5 sehingga menghasilkan jumlah generasi yang optimal terdapat pada generasi ke-150. Pada pengujian jumlah generasi individu pengawas terlihat grafik terus mengalami peningkatan hingga generasi ke-400 mencapai rata-rata nilai *fitness* tertinggi dan mulai terlihat stabil. Individu dengan jumlah generasi yang lebih besar cenderung menghasilkan solusi yang lebih optimal. Namun hal ini juga akan memengaruhi waktu komputasi sehingga proses akan memakan waktu yang lama.



Gambar 6.5 Grafik pengujian jumlah generasi mata pelajaran



Gambar 6.6 Grafik pengujian jumlah generasi individu pengawas

6.4 Analisis Global

Berdasarkan tiga pengujian yang telah dilakukan sebelumnya yaitu pengujian ukuran populasi, kombinasi nilai *crossover rate* dan *mutation rate* serta pengujian jumlah generasi diperoleh ukuran populasi sebesar 150 dengan kombinasi nilai *crossover rate* 0.7 dan *mutation rate* 0.3 pada generasi 200 untuk menghasilkan individu mata pelajaran yang optimal. Sedangkan parameter untuk individu pengawas yang optimal diperoleh pada ukuran populasi 180 dengan kombinasi nilai *crossover rate* 0.6 dan *mutation rate* 0.4 pada generasi 400. Individu mata pelajaran menghasilkan nilai *fitness* yang optimal sebesar 0.1361 yang ditunjukkan pada Tabel 6.5 dan individu pengawas memiliki nilai *fitness* yang optimal sebesar 0.00642 yang ditunjukkan pada Tabel 6.6.

Tabel 6.5 Detail pelanggaran *constraint* individu mata pelajaran

No	Constraint	Penalti Sistem	Penalti Manual
1	Terdapat bentrok mata pelajaran dalam satu hari	5.772	0
2	Terdapat mata pelajaran dalam satu hari yang memiliki bobot > 6.29	0.286	0
3	Terdapat mata pelajaran sesi 2 memiliki bobot lebih besar dari pada sesi 1	0.286	0
4	Terdapat mata pelajaran ujian nasional pada sesi 2	0	0
Jumlah penalti		6.344	0
Fitness		0.1361	1

Tabel 6.6 Detail pelanggaran *constraint* individu pengawas

No	<i>Constraint</i>	Penalti Sistem	Penalti Manual
1	Terdapat bentrok jadwal mengawas pada tiap sesi	1529.5	0
2	Pengawas tidak mengawas sesuai mata pelajaran yang dikuasai	10.65	0
3	Terdapat guru senior mengawas di sesi ke 2	16.0	0
Jumlah penalti		1556.15	0
<i>Fitness</i>		0.00642	1

Untuk menentukan kualitas oleh algoritme genetika dapat diukur dengan nilai *fitness* yang telah dihasilkan. Semakin besar nilai *fitness* yang dihasilkan menunjukkan solusi tersebut semakin baik dan begitu juga sebaliknya. Berdasarkan parameter algoritme genetika optimal yang telah diperoleh nilai *fitness* untuk individu mata pelajaran sebesar 0.1361 dan 0.00642 untuk individu pengawas yang masih belum optimal. Hal ini disebabkan masih terdapat pelanggaran *constraint* yang telah ditentukan pada kedua individu. Banyak dari data yang ada menjadi salah satu penyebabnya. Individu awal yang dibangkitkan secara acak juga tidak menjamin susunan individu yang baik karena terdapat beberapa data yang sama muncul berulang kali. Kemampuan algoritme genetika untuk melakukan eksplorasi juga kurang luas karena beberapa individu memiliki tingkat kemiripan yang tinggi antara satu sama lain. Pelanggaran terhadap *constraint* yang ditentukan masih terjadi karena pencarian solusi yang kurang luas dan hal ini juga akan berdampak pada lamanya waktu komputasi.

Algoritme genetika mampu diterapkan pada penjadwalan ujian akhir semester dan untuk mendapatkan solusi yang lebih optimal dibutuhkan ukuran populasi dan jumlah generasi yang besar. Hal tersebut juga berpengaruh terhadap waktu yang dibutuhkan untuk melakukan komputasi. Namun penggunaan algoritme genetika dalam penyusunan jadwal ujian akhir semester menjadi lebih cepat dilakukan apabila dibandingkan dengan penyusunan secara manual yang membutuhkan waktu 2-3 hari. Hasil dari jadwal mata pelajaran dan jadwal pengawas yang lebih detail dapat dilihat pada LAMPIRAN C dan LAMPIRAN D.

BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan mengenai optimasi pada permasalahan penjadwalan ujian akhir semester di SMAN 5 Malang dengan menggunakan algoritme genetika diperoleh kesimpulan sebagai berikut:

1. Parameter yang digunakan pada penelitian ini yaitu ukuran populasi, jumlah generasi serta kombinasi nilai *crossover rate* dan *mutation rate*. Pengujian untuk masing-masing parameter dilakukan dengan tujuan untuk menemukan parameter yang tepat pada algoritme genetika. Dari hasil pengujian ditemukan jumlah populasi optimal untuk individu mata pelajaran sejumlah 180 sedangkan untuk individu pengawas sejumlah 150. Pada pengujian kombinasi nilai *crossover rate* dan *mutation rate* ditemukan kombinasi nilai optimal untuk individu mata pelajaran adalah *crossover rate* sebesar 0.7 dan *mutation rate* sebesar 0.3 dan untuk individu pengawas adalah nilai *crossover rate* sebesar 0.6 dan *mutation rate* sebesar 0.4. Jumlah generasi optimal pada individu mata pelajaran adalah 200 generasi sedangkan pada individu pengawas adalah 400 generasi. Semakin besar ukuran populasi akan memperbesar ruang pencarian solusi optimal. Semakin besar jumlah generasi juga akan menghasilkan solusi yang cenderung optimal namun akan berpengaruh pula pada waktu komputasi. Nilai *crossover rate* yang lebih besar menyebabkan *offspring crossover* lebih besar sehingga ruang eksplorasi algoritme genetika tidak luas dan menghasilkan *offspring* yang mirip dengan induknya.
2. Nilai *fitness* optimal yang diperoleh pada ukuran populasi mata pelajaran sebesar 180 adalah 0.144613584 sedangkan nilai *fitness* optimal untuk populasi pengawas sejumlah 150 populasi adalah 0.004503. Pada pengujian kombinasi nilai *crossover rate* dan *mutation rate* pada populasi mata pelajaran diperoleh nilai *fitness* optimal sebesar 0.14083 sedangkan kombinasi nilai *crossover rate* pada populasi pengawas diperoleh *fitness* sebesar 0.00329. Pengujian jumlah generasi mata pelajaran sebanyak 150 menghasilkan nilai *fitness* yang optimal sebesar 0.129539 dan untuk jumlah generasi pengawas sebanyak 150 generasi menghasilkan nilai *fitness* yang optimal sebesar 0.003059207. Dari nilai *fitness* dapat dilihat bahwa jadwal yang dihasilkan masih belum optimal.

7.2 Saran

Saran yang dapat diberikan berdasarkan hasil penelitian yang telah dilakukan adalah penggunaan metode *crossover* serta mutasi yang berbeda untuk mendapatkan variasi solusi yang lain serta bentuk representasi kromosom yang berbeda dapat dilakukan untuk penelitian selanjutnya. Perbandingan beberapa metode seleksi seperti *elitism selection* dengan *tournament selection* ataupun *roulette wheel selection* dapat dilakukan untuk mendapatkan solusi yang lebih optimal.

DAFTAR PUSTAKA

- Black, P., 2001. *Dictionary of Algorithms and Data Structures*. [Online] Available at: <https://xlinux.nist.gov/dads/HTML/optimization.html> [Accessed 27 Februari 2018].
- Colorni, A., Dorigo, M. & Maniezzo, V., 1992. A Genetic Algorithm to Solve the Timetable Problem. *Technical Report 90-060 revised, Politecnico di Milano, Italy*.
- Damayanti, C. P., Putri, R. R. M. & Fauzi, M. A., 2017. Implementasi Algoritma Genetika Untuk Penjadwalan Customer Service (Studi Kasus: Biro Perjalanan Kangaroo). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 456-465 .
- Gen, M. & Cheng, R., 1999. *Genetic Algorithms*. New York: John Wiley & Sons, Inc.
- Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization, and Machine*. Reading: Addison-Wesley.
- Holland, J. H., 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press.
- Jain, A., Jain, D. S. & Chande, D. P., 2010. Formulation of Genetic Algorithm to Generate Good Quality Course Timetable. *International Journal of Innovation, Management and Technology*, Volume 1, pp. 249-251.
- Jong, K. A. D. & Spears, W. M., 1992. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, Volume 5, pp. 1-26.
- Lukas, S., Aribowo, A. & Muchri, M., 2009. Genetic Algorithm and Heuristic Search For Solving Timetable Problem. *2009 Second International Conference on the Applications of Digital Information and Web Technologies* , pp. 629-633.
- Magalhães-Mendes, J., 2011. A Two-level Genetic Algorithm for the Multi-Mode Resource-Constrained Project Scheduling Problem. *International Journal of Systems Applications, Engineering & Development*, pp. 274-278.
- Mahmudy, W. F., 2013. *Algoritma Evolusi (Modul Kuliah Semester Ganjil 2013-2014)*. Malang: Program Teknologi Informasi dan Ilmu (PTIIK).
- Mittal, D., Doshi, H., Sunasra, M. & Nagpure, R., 2015. Automatic Timetable Generation using Genetic Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(2), pp. 246-248.
- Promcharoen, S., Wirayut, P. & Tanhan, P., 2009. Genetic Algorithm for Exam Proctor. *ICROS-SICE International Joint Conference 2009*, pp. 2903-2906.
- Ram, M. & Davim, J. P., 2017. *Soft Computing Techniques and Applications in Mechanical*. Hersey: IGI Global.

- Rana, S., Howe, A. E., Whitley, L. D. & Mathias, K., 1996. Comparing Heuristic, Evolutionary and Local Search Approaches to Scheduling. *Proceedings of Third Conference on AI Planning System*, pp. 174-181.
- Saputro, H. A., Mahmudy, W. F. & Dewi, C., 2015. Implementasi Algoritma Genetika untuk Optimasi Penggunaan Lahan Pertanian. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, Volume 5.
- Schaerf, A., 1999. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2), pp. 87-127.
- Umbarkar, A. & Sheth, P., 2015. Crossover Operators in Genetic Algorithms: A Review. *ICTACT JOURNAL ON SOFT COMPUTING*, 6(1), pp. 1083-1092.
- Woods, D. & Trenaman, A., 1999. Simultaneous Satisfaction of Hard and Soft Timetable.
- Yuan, J. et al., 2013. A novel genetic probability decoding (GPD) algorithm for the FEC code in optical. *Optik - International Journal for Light and Electron Optics*, 124(15), pp. 1986-1989.

